

Dxx 解码器 Api 文档 (v2.2) -2.02

Dxx 系列解码器接口

欢迎您使用 Device HTTP API!

Device HTTP API 是为本型号网络解码和软件开发编程接口。Device HTTP API 旨在通过简单的 HTTP 请求/响应机制，实现对设备的各种功能控制。软件开发者利用 Device HTTP API，可以达到远程管理和控制设备的目的。

API 的每一个功能，对应有一个 HTTP 请求路径。在当前版本的 API 中，这个路径的格式是：
`http://[:]/api/[模块名称]/[方法] (http:ip:port/api/xxx)`

本文档将列出每一个 API 功能对应的文件名(filename)及其功能描述。同时，在每个 API 文档中，会有一个测试接口，您可以根据请求参数的描述，在测试接口中填写您需要进行测试的请求参数，执行测试，可以在线验证 API 功能，十分方便。

API 的请求遵循标准的 HTTP 协议，开发人员可以使用任何符合 HTTP 标准的工具、软件开发库来完成 API 调用（例如 Web 浏览器，JavaScript 库，C/C++的 HTTP 协议库，JAVA，.NET 等）。参数的提交根据不同 API 的接口要求，可能为 POST 或 GET，或者两者均可，具体请参见 API 说明。API 的响应为 JSON 格式，JSON 对象中的每一个参数的意义，请参见每一个 API 的文档说明。

我们尽可能保持产品 API 接口的一致，但不可排除因产品特性差异而导致不同产品的 API 接口会存在部分差异。因此，请开发人员按照官方指导，区别和处理这些细节差异。

目录

- 一、[调用接口说明](#)
 - 二、[鉴权](#)
 - 三、[示例](#)
 - 四、[错误码](#)
 - 五、[INFO 模块](#)
 - 六、[USER 模块](#)
 - 七、[OVERLAY 模块](#)
 - 八、[SOURCE 模块](#)
 - 九、[LAYOUT 模块](#)
 - 十、[OUTPUT 模块](#)
 - 十一、[NETWORK 模块](#)
 - 十二、[SYSTIME 模块](#)
 - 十三、[SYSTEM 模块](#)
 - 十四、[FIRMWARE 模块](#)
 - 十五、[PREVIEW 模块](#)
 - 十六、[NDI-Discovery](#)
-

（一）调用接口说明

统一接口前缀：

Base URLs: http:ip:port/api/xxx

所有接口的返回形式都统一为：

- 正常返回 `{"result": "ok", "data": 某种类型的数据, 比如字符串、数值、数组、对象等等, 可能没有这个字段}` 无 data 字段返回示例: `{"result": "ok"}`
- 错误返回 `{"result": "error", "msg": "errCode"}`

（二）鉴权

- header
 - Parameter Name: `**app**`, in: header.

（三）示例

（1）API 认证原理

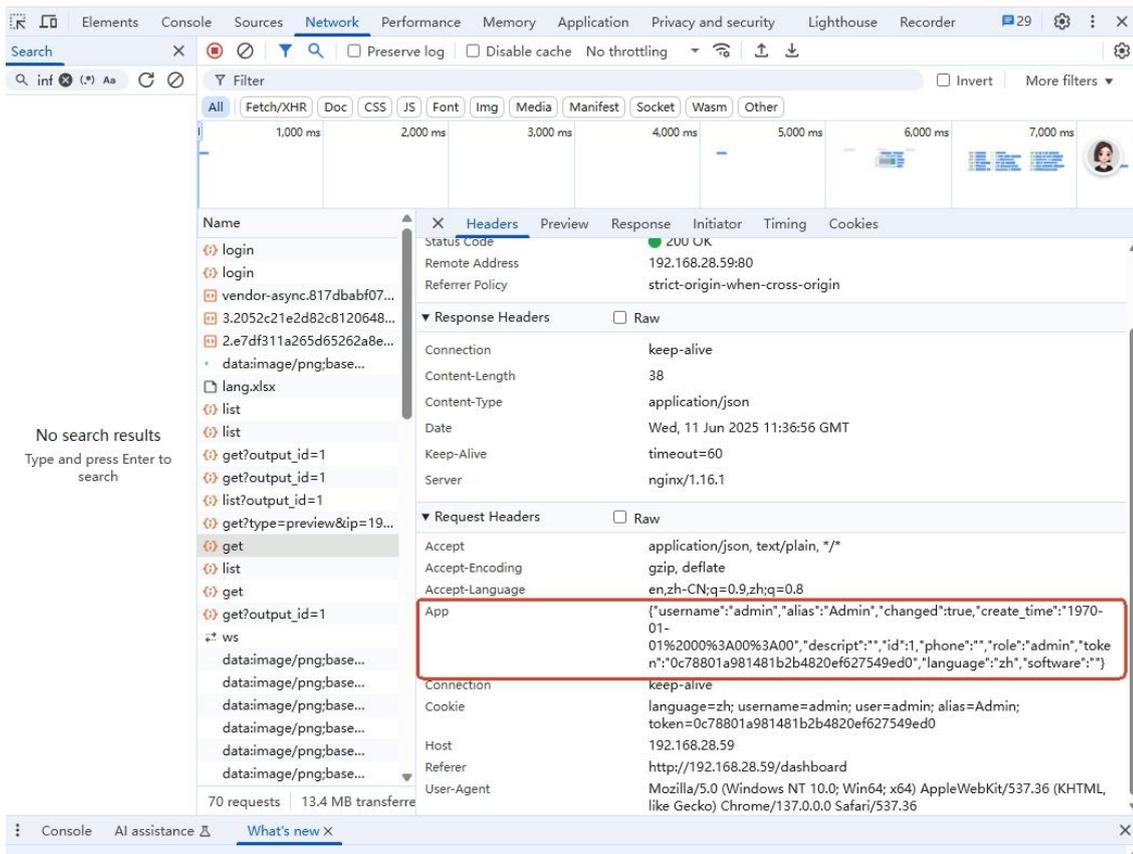
（2）登录获取令牌

当调用登录 API（通过 GET/POST 方式）时，响应会包含一个 token。该令牌将作为后续 API 请求的认证凭证。

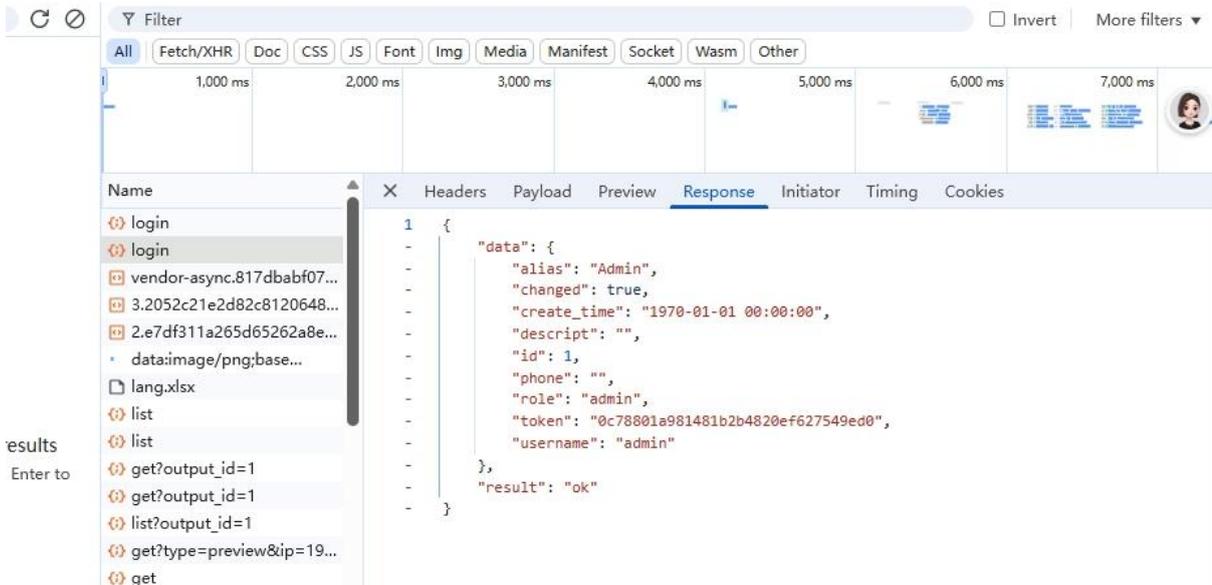
（3）在请求中使用令牌

对于其他任何 API 端点，必须在请求头中添加令牌，格式如下（示例）：

```
{"username": "admin", "alias": "admin", "changed": true, "role": "admin", "token": "3515d2e8d3615fc68c84f00b361c5d8c", "language": "en", "software": "true"}.
```



可在浏览器中按 F12 打开开发者工具获取相关应用。



令牌的作用是向服务器证明已通过认证，原理类似 Web 图形界面通过 Cookie 维持会话状态。

(4) 本地演示步骤 (含 Postman 操作)

(5) 步骤 1: 登录获取令牌

- a. 打开 Postman, 新建 POST 请求。
- b. 请求地址: <https://192.168.28.30/api/users/login>
- c. 请求体 (JSON 格式): {"username": "admin", "password": "your-device-password" }

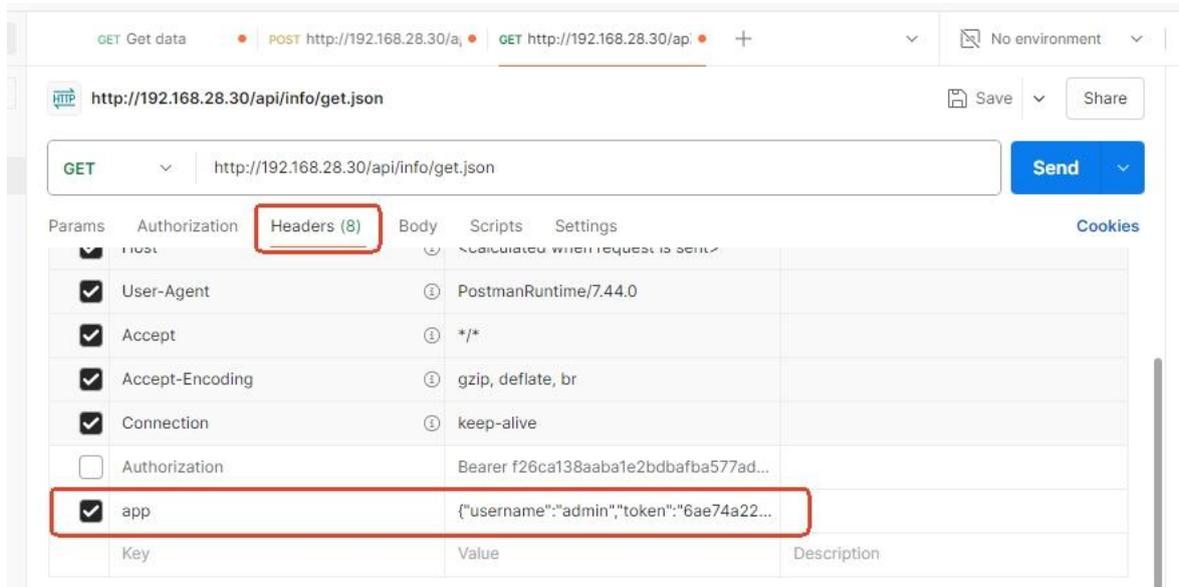
(注: 若设备未修改密码, 默认密码为 admin。)
- d. 发送请求: 点击 Send, 从响应中提取 token 字段 (如 3515d2e8d3615fc68c84f00b361c5d8c)。

(6) 步骤 2: 在请求中添加认证头部

- a. 新建目标 API 请求 (如获取信息的 info/get.json)。
- b. 设置请求头:
- e. 切换至 Headers 标签页, 添加以下键值对:
- f. Key: app
- g. Value: {"username":"admin","token":"your-obtained-token"}
(注: 将 your-obtained-token 替换为步骤 1 中实际获取的令牌。)
- c. 发送请求: 点击 Send, 服务器通过令牌验证身份后返回数据。

(7) 浏览器开发者工具获取令牌 (备选方法)

- a. 浏览器按 F12 打开开发者工具, 切换至 Network 标签页。
- b. 执行登录操作, 找到登录请求 (如 login.json), 在响应体或头部中查看 token 字段。



(8) 故障排除提示

- 密码验证：确保设备密码正确（未修改时默认密码为 admin）。
- 令牌有效性：检查令牌是否复制正确，且未过期（过期需重新登录获取）。
- 请求头格式：确认 app 头部的 JSON 格式正确，令牌字段无拼写错误（如 token 需与服务器要求一致）。

(四) 错误码

错误码	错误描述
0	请求成功
1	参数错误
2	鉴权错误
100	输出未找到
101	视频源未找到
102	音频源未找到
103	输出接口未找到
104	布局模板未找到
105	预设未找到
106	输出接口未找到
107	布局超出范围
200	组未找到
201	流未找到
202	流不存在
203	NDI 服务器重复
204	流重复
205	不支持的协议
206	分组已存在
207	缓存超出最大限制
300	用户未找到
301	密码不匹配
302	用户名或密码错误
400	未 ping 通
401	ERRMODIFYNET
402	网络配置错误
403	路由重复
404	路由未找到
405	端口已被占用
500	名称重复
501	正在使用
502	保存配置失败
503	时间设置失败

504	无效的 VO 输出
505	混音器只支持四路音频
506	目标已存在
507	文件不存在
508	PTZ 设置失败
509	设备性能不足
1000	公共错误码

二、INFO 模块

GET /info/get

1. 请求参数

名称	位置	类型	必选	说明
app	header	string	是	none

返回示例

```
{
  "data": {
    "device_name": "xxx",
    "firmware_version": "xxx",
    "hardware_version": "xxx",
    "serial_number": "xxx",
    "software_version": "2.0.0"
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» device_name	string	true	none		设备名称
»» firmware_version	string	true	none		固件版本

»» hardware_version	string	true	none		硬件版本
»» serial_number	string	true	none		设备序列号
»» software_version	string	true	none		软件版本
» result	string	true	none		none

三、USER 模块

(一) POST /users/login

POST /users/login

用户登录

Body 请求参数

```
{
  "username": "test",
  "password": "Admin123"
}
```

1. 请求参数

名称	位置	类型	必选	说明
username	query	string	否	用户名
password	query	string	否	密码
app	header	string	是	none
body	body	object	否	none

返回示例

```
{
  "data": {
    "alias": "ad nostrud fugiat veniam",
    "changed": true,
    "create_time": "2006-02-28 20:50:14",
    "descript": "reprehenderit culpa dolore voluptate est",
    "id": 4,
    "phone": "18176237335",
    "role": "admin",
    "token": "token",
  }
}
```

```
"username": "admin"
},
"result": "velit elit id"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» alias	string	true	none		别名
»» changed	boolean	true	none		是否修改过密码
»» create_time	string	true	none		创建时间
»» descript	string	true	none		描述
»» id	integer	true	none		id, 唯一值
»» phone	string	true	none		联系电话
»» role	string	true	none		权限
»» token	string	true	none		验证 token
»» username	string	true	none		账号名称
» result	string	true	none		none

(二) POST /users/logout

POST /users/logout

退出登录

Body 请求参数

```
{
  "token": "string"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» token	body	string	是	鉴权 token	none

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) GET /users/list

GET /users/list

获取用户列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "count": 1,
  "list": [
    {
      "alias": "admin",
      "create_time": "1715070431000",
      "descript": "初始密码 admin，登陆时需强制修改密码",
      "id": 1,
      "phone": "",
      "role": "admin",
      "token": "caf27900204ebde13710feef58a2208c",
      "username": "admin"
    }
  ]
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» count	integer	true	none		none

» list	[object]	true	none		none
»» alias	string	false	none		别名
»» create_time	string	false	none		创建时间戳 (ms)
»» descript	string	false	none		详细描述
»» id	integer	false	none		用户 id
»» phone	string	false	none		电话号码
»» role	string	false	none		角色
»» token	string	false	none		token 值
»» username	string	false	none		用户名

(四) POST /users/add

POST /users/add

添加用户

Body 请求参数

```
{
  "username": "admin",
  "password": "admin",
  "alias": "test1",
  "role": "user",
  "phone": "",
  "descript": ""
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	string	否		none
» username	body	string	是		用户名称
» password	body	string	是		密码
» alias	body	string	是		别名
» role	body	string	是		角色
» phone	body	string	是		联系电话
» descript	body	string	是		描述

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(五) POST /users/remove

POST /users/remove

删除用户

Body 请求参数

```
{
  "id": 2
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	integer	是		用户 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /users/modifyPwd

POST /users/modify

修改密码

Body 请求参数

```
{
  "username": "admin",
  "password": "Admin123",
  "passwordAgain": "Admin123"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» username	body	string	是		用户名称
» password	body	string	是		密码
» passwordAgain	body	string	是		密码

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

四、OVERLAY 模块

(一) GET /overlay/list

GET /overlay/list

获取叠加模板列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否		none
app	header	string	是		none

返回示例

```

{
  "data": [
    {
      "background": "transparent",
      "editing": true,
      "id": 1,
      "img_path": "./data/decoder_config/overlay/preview/overlay_1.png",
      "modified": false,
      "name": "template_1",
      "selected": false,
      "status": 0
    }
  ],
  "result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	none		none
»» background	string	false	none		背景颜色
»» editing	boolean	false	none		是否编辑
»» id	integer	false	none		id
»» img_path	string	false	none		图片地址
»» modified	boolean	false	none		是否修改
»» name	string	false	none		名称
»» selected	boolean	false	none		是否选中
»» status	integer	false	none		状态
» result	string	true	none		none

(二) POST /overlay/select

POST /overlay/select

选择（勾选）某一模板

Body 请求参数

```
{
  "overlay_id": 1,
  "output_id": 0
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» overlay_id	body	integer	是		叠加模板 id
» output_id	body	integer	是		输出 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

五、SOURCE 模块

(一) POST /source/groups/list

POST /source/groups/list

获取所有视频源组

Body 请求参数

```
{
  "is_need_stream": true
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
»	body	boolean	否	是否携带流信	用户控制返回

is_need_stream				息返回	组列表的同时 是否返回流信 息
------------------	--	--	--	-----	-----------------------

返回示例

```
{
  "data": [
    {
      "id": "2ed1f7150cf235e8662d73ac58b925b1",
      "name": "NDI",
      "streams": [
        {
          "address": "192.168.36.238",
          "application": "",
          "audio_sync_compst": 0,
          "bandwidth": -1,
          "buffer": "live:0:0:0:0:0",
          "connect_speed": 0,
          "connection_mode": "",
          "enable": 0,
          "encryption": "",
          "id": "0f79952dbc00505b2331b0bdb0c8ca0a",
          "latency": -1,
          "listener_port": -1,
          "name": "N60_238 (HX)",
          "ndi_name": "N60_238 (HX)",
          "passphrase": "",
          "password": "",
          "payload_size": -1,
          "preview_mode": "Auto",
          "rtmp_server_port": -1,
          "sdp_content": "",
          "security": "",
          "session": "public",
          "show_advanced": "",
          "srt_id": "",
          "status": 2,

```

```
"token": "",
"trans_mode": "tcp",
"type": "ndi",
"url": "192.168.36.238:5964",
"user": ""
},
{
"address": "192.168.36.238",
"application": "",
"audio_sync_compst": 0,
"bandwidth": -1,
"buffer": "live:0:0:0:0:0",
"connect_speed": 0,
"connection_mode": "",
"enable": 0,
"encryption": "",
"id": "e893d03a94326b981fd3aa276f90183a",
"latency": -1,
"listener_port": -1,
"name": "N60_238 (HB)",
"ndi_name": "N60_238 (HB)",
"passphrase": "",
"password": "",
"payload_size": -1,
"preview_mode": "Auto",
"rtmp_server_port": -1,
"sdp_content": "",
"security": "",
"session": "public",
"show_advanced": "",
"srt_id": "",
"status": 2,
"token": "",
"trans_mode": "tcp",
"type": "ndi",
"url": "192.168.36.238:5965",
```

```
    "user": ""
  }
]
},
{
  "id": "0093969758ac42c99245e89f9108ad08",
  "name": "Other",
  "streams": [
    {
      "address": "",
      "application": "",
      "audio_sync_compst": 0,
      "bandwidth": 0,
      "buffer": "live:200:200:500:0:0",
      "connect_speed": 0,
      "connection_mode": "",
      "enable": 0,
      "encryption": "",
      "id": "03db9d3284fd709f6bdd099c9a5f8778",
      "latency": 0,
      "listener_port": 0,
      "name": "RTMP",
      "ndi_name": "",
      "passphrase": "",
      "password": "",
      "payload_size": 0,
      "preview_mode": "Auto",
      "rtmp_server_port": 0,
      "sdp_content": "",
      "security": "",
      "session": "",
      "show_advanced": "",
      "srt_id": "",
      "status": 0,
      "token": "",
      "trans_mode": "",
```

```
"type": "rtmp",
"url": "rtmp://192.168.0.73:1935/live/vwv",
"user": ""
},
{
  "address": "192.168.36.236",
  "application": "",
  "audio_sync_compst": 0,
  "bandwidth": 25,
  "buffer": "live:200:200:500:0:0",
  "connect_speed": 0,
  "connection_mode": "Caller",
  "enable": 0,
  "encryption": "none",
  "id": "834fd00666b91015b23ee336bc2d605a",
  "latency": 125,
  "listener_port": 1025,
  "name": "SRT_Caller",
  "ndi_name": "",
  "passphrase": "",
  "password": "",
  "payload_size": 1316,
  "preview_mode": "Auto",
  "rtmp_server_port": 0,
  "sdp_content": "",
  "security": "",
  "session": "",
  "show_advanced": "0",
  "srt_id": "",
  "status": 0,
  "token": "",
  "trans_mode": "",
  "type": "srt",
  "url": "srt://192.168.36.236:1025",
  "user": ""
},
```

```
{
  "address": "",
  "application": "",
  "audio_sync_compst": 0,
  "bandwidth": 0,
  "buffer": "live:200:200:500:0:0",
  "connect_speed": 0,
  "connection_mode": "",
  "enable": 0,
  "encryption": "",
  "id": "f63dda56f9891ed3b4a5677cf040a4f8",
  "latency": 0,
  "listener_port": 0,
  "name": "RTMP",
  "ndi_name": "",
  "passphrase": "",
  "password": "",
  "payload_size": 0,
  "preview_mode": "Auto",
  "rtmp_server_port": 0,
  "sdp_content": "",
  "security": "",
  "session": "",
  "show_advanced": "",
  "srt_id": "",
  "status": 0,
  "token": "",
  "trans_mode": "",
  "type": "rtmp",
  "url": "rtmp://192.168.0.73:1935/live/vw",
  "user": ""
}
]
}
```

```
"result": "ok"
}
```

2. 返回数据结构

状态码 0

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» id	string	true	none		组 id
»» name	string	true	none		组名称
»» streams	\[object]	true	none		none
»»» address	string	true	none		公网 ip
»»» application	string	true	none		rtmp_server 应用名称
»»» audio_sync_compst	integer	true	none		保留字段
»»» bandwidth	integer	true	none		srt 恢复带宽开 销
»»» buffer	string	true	none		缓存时间
»»» connect_speed	integer	true	none		保留字段
»»» connection_mode	string	true	none		srt 连接模式 Rendezvous or Listener or Caller
»»» enable	integer	true	none		保留字段
»»» encryption	string	true	none		srt 加密方式
»»» id	string	true	none		流 id,唯一值
»»» latency	integer	true	none		Srt 数据包延迟
»»» listener_port	integer	true	none		端口
»»» name	string	true	none		流名字
»»» ndi_name	string	true	none		NDI 名称
»»» passphrase	string	true	none		srt 字符密钥
»»» password	string	true	none		连接鉴权密码
»»» payload_size	integer	true	none		srt 加载大小
»»» preview_mode	string	true	none		加 载 模 式 Auto (根据输 出分辨率自动

					选择主码流或子码流) Full(主码流) Preview(子码流)
»»» rtmp_server_port	integer	true	none		rtmp_server 端口
»»» sdp_content	string	true	none		rtp 协议 sdp 内容
»»» security	string	true	none		rtmp_server 验证方式 none or token
»»» session	string	true	none		rtmp_server 会话
»»» show_advanced	string	true	none		是否显示高级 设置
»»» srt_id	string	true	none		srt 协议的流 id
»»» status	integer	true	none		拉流状态 0 未 连接 1 正在连 接 2 连接成功 3 连接失败 4 性能受限制, 停 止拉流
»»» token	string	true	none		rtmp_server 鉴权 token
»»» trans_mode	string	true	none		rtsp 连接模式 udp/tcp
»»» type	string	true	none		流类型
»»» url	string	true	none		流 url
»»» user	string	true	none		连接鉴权账号
»»» connect_speed	integer	true	none		连接超时时间
»»» audio_sync_compst	integer	true	none		音频同步补偿 (ms)
» result	string	true	none		none

(二) POST /source/groups/add

POST /source/groups/add

添加一组

Body 请求参数

```
{
  "name": "group1"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» name	body	string	是		添加组的名称

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) POST /source/groups/streams/add

POST /source/groups/streams/add

添加流

Body 请求参数

```
//rtsp 添加
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f", //所属组
  "name": "rtsp1", //名称 --非空
  "type": "rtsp", //类型
  "url": "rtsp://192.168.36.222:554/ch01", //拉流地址
  "buffer": "live:0:0:0:0:0", //缓存
  "trans_mode": "tcp", //rtsp 传输模式 tcp or udp
}
```

```
"connect_speed":5000, //连接速度
"audio_sync_compst":0, //音频同步补偿
"user": "", //鉴权账号 --可为空
"password": "" //鉴权密码--可为空
}

//rtmp 添加
{
  "group_id": "d054f1d80ed71583392fada325a02823", //所属组
  "name": "rtmp1", //名称 --非空
  "type": "rtmp", //类型
  "url": "rtmp://192.168.0.73:1935/live/vw", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "connect_speed":5000, //连接速度
  "audio_sync_compst":0, //音频同步补偿
  "user": "", //鉴权账户
  "password": "" //鉴权密码
}

//rtmps 添加
{
  "group_id": "d054f1d80ed71583392fada325a02823", //所属组
  "name": "rtmps", //名称 --非空
  "type": "rtmps", //类型
  "url": "rtmps://192.168.0.73:1935/live/vw", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "connect_speed":5000, //连接速度
  "audio_sync_compst":0, //音频同步补偿
  "user": "", //鉴权账户
  "password": "" //鉴权密码
}

//srt_listener
{
  "group_id": "8eca38e84020aba241ead45094c7dd24", //所属组
  "name": "srt_listener", //名称
```

```
"type": "srt", //类型
"url": "srt://:1025", //srt listener 模式没有拉流 url. 固定传: srt://:port
"buffer": "live:0:0:0:0", //缓存
"connection_mode": "Listener", //srt 连接模式 Rendezvous or Listener or Caller
"listener_port": 1025, //srt 连接端口
"latency": 125, //IP 生存时间
"connect_speed": 5000, //连接速度
"audio_sync_compst": 0, //音频同步补偿
"show_advanced": "1", //是否显示高级设置
"srt_id": "111", //srt id
"encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
"passphrase": "", //加密密码
"bandwidth": 25, //带宽开销
"payload_size": 1316 //载荷大小
}

//srt_rendezvous
{
  "group_id": "52005872cd91b13767660d196c463879", //所属组
  "name": "srt3", //名称
  "type": "srt", //类型
  "url": "srt://192.168.36.51:1025", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "connection_mode": "Rendezvous", //srt 连接模式 Rendezvous or Listener or Caller
  "address": "192.168.36.51", //公网 IP
  "listener_port": 1025, //srt 连接端口
  "latency": 125, //IP 生存时间
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "show_advanced": "1", //是否显示高级设置
  "srt_id": "111", //srt id
  "encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
  "passphrase": "", //加密密码
  "bandwidth": 25, //带宽开销
  "payload_size": 1316 //载荷大小
}
```

```
//srt_caller
{
  "group_id": "8eca38e84020aba241ead45094c7dd24", //所属组
  "name": "srt_caller", //名称
  "type": "srt", //类型
  "url": "srt://192.168.36.59:1025", //拉流地址
  "buffer": "live:0:0:0:0:0", //缓存
  "connection_mode": "Caller", //srt 连接模式 Rendezvous or Listener or Caller
  "address": "192.168.36.51", //公网 IP,需要跟 url 中的保持一致
  "listener_port": 1025, //srt 连接端口
  "latency": 125, //IP 生存时间
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "show_advanced": "1", //是否显示高级
  "srt_id": "111", //srt id
  "encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
  "passphrase": "", //加密密码
  "bandwidth": 25, //带宽开销
  "payload_size": 1316 //载荷大小
}

//rtmp_server
{
  "group_id": "8eca38e84020aba241ead45094c7dd24", //所属组
  "name": "rtmp server1", //名称
  "type": "rtmp_server", //类型
  "url": "rtmp://192.168.36.59:1935/live/xxx/11111", //拉流地址
  "buffer": "live:0:0:0:0:0", //缓存
  "application": "live", //rtmp 应用名称
  "session": "11111", //会话名称
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "security": "token", //验证方式 "" or "token"
  "token": "xxx", //token 内容
  "enable": 1, //默认 1
}
```

```
"rtmp_server_port":1935 //rtmp 端口 大于 1025 小于 65535
}

//rtp
{
  "group_id": "e64fbc915a0c34c85cb83bac1e1650fe", //所属组
  "name": "rtp_no_audio", //名称
  "type": "rtp", //类型
  "connect_speed":5000, //连接速度
  "audio_sync_compst":0, //音频同步补偿
  "buffer": "live:200:200:500:0:0", //缓存
  "sdp_content": "SDP:\nv=0\no=- 0 0 IN IP4 127.0.0.1\ns=No Name\nc=IN IP4 192.168.36.197\nt=0
0\na=tool:libavformat 61.1.100\nm=video 1234 RTP/AVP 96\na=rtpmap:96 H264/90000\na=fmtp:96
packetization-mode=1" //sdp 内容
}

//udp
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f", //所属组
  "name": "udp", //名称 --非空
  "type": "udp", //类型 --非空
  "connect_speed":5000, //连接速度
  "audio_sync_compst":0, //音频同步补偿
  "url": "rtsp://192.168.36.222:554/ch01", //拉流地址
  "buffer": "live:0:0:0:0:0" //缓存
}

//http
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f", //所属组
  "name": "http", //名称 --非空
  "type": "http", //类型 --非空
  "connect_speed":5000, //连接速度
  "audio_sync_compst":0, //音频同步补偿
  "url": "http://192.168.36.222:554/ch01", //拉流地址
  "buffer": "live:0:0:0:0:0" //缓存
```

```

}
//https
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f", //所属组
  "name": "https", //名称 --非空
  "type": "https", //类型 --非空
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "url": "https://192.168.36.222:554/ch01", //拉流地址
  "buffer": "live:0:0:0:0" //缓存
}

//ndi
{
  "group_id": "c719a3c5803d9434050f6ede09d29215", //所属组
  "type": "ndi", //流类型
  "data": [
    {
      "channel": "HB", //HB or HX
      "group_name": "public", //group
      "ip": "192.168.36.238", //ip
      "name": "N60_238 ", //名称
      "ndi_name": "N60_238 (HB)", //ndi 名称
      "connect_speed": 5000, //连接速度
      "audio_sync_compst": 0, //音频同步补偿
      "port": 5965, //端口
      "url": "192.168.36.238:5965" //url
    }
  ]
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» group_id	body	string	是		所属组 ID

» name	body	string	是		名称
» type	body	string	是		流 类 型 支 持 rtmp/rtmps/rtsp/udp/http/https/srt/rtp/ndi
» url	body	string	是		none
» buffer	body	string	是		none
» address	body	string	是		公网 ip
» application	body	string	是		rtmp_server 应用名称
» audio_sync_compst	body	integer	否		保留字段
» bandwidth	body	integer	否		srt 恢复带宽开销
» connect_speed	body	integer	否		保留字段
» connection_mode	body	string	否		srt 连接模式 Rendezvous or Listener or Caller
» enable	body	integer	否		保留字段
» encryption	body	string	否		srt 加密方式
» id	body	string	否		流 id,唯一值
» latency	body	integer	否		Srt 数据包延迟
» listener_port	body	integer	否		端口
» ndi_name	body	string	否		NDI 名称
» passphrase	body	string	否		srt 字符密钥
» password	body	string	否		连接鉴权密码
» payload_size	body	integer	否		srt 加载大小
» preview_mode	body	string	否		加载模式 Auto (根据输出分辨率自动选择主码流或子码流) Full(主码流) Preview(子码流)
» rtmp_serverport	body	integer	否		rtmp_server 端口
» sdp_content	body	string	否		rtp 协议 sdp 内容
» security	body	string	否		rtmp_server 验证方式 none or token
» session	body	string	否		rtmp_server 会话
» show_advanced	body	string	否		是否显示高级设置
» srt_id	body	string	否		srt 协议的流 id
» status	body	integer	否		拉流状态 0 未连接 1 正在连接 2 连接成功 3 连接失败 4 性能受限制, 停止拉流
» token	body	string	否		rtmp_server 鉴权 token
» trans_mode	body	string	否		rtsp 连接模式 udp/tcp

| » user | body | string | 否 | | 连接鉴权账号

\

» connect_speed	integer	true	none		连接超时时间
-----------------	---------	------	------	--	--------

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(四) POST source/groups/rename

POST /source/groups/rename

修改指定组名称

Body 请求参数

```
{  
  "group_id": 1,  
  "name": "new_group1"  
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» group_id	body	string	是		组 id
» name	body	string	是		新修改的组名称

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
----	----	----	----	-----	----

» result	string	true	none		none
----------	--------	------	------	--	------

(五) POST source/groups/remove

POST /source/groups/remove

删除某一组

Body 请求参数

```
{
  "group_id": "string"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» group_id	body	string	是		组 id

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /source/groups/streams/modify

POST /source/groups/streams/modify

修改某一视频流参数，可以根据协议类型传递对应参数。

Body 请求参数

```
//修改 rtmp
{
  "stream_id": "d054f1d80ed71583392fada325a02823", //stream_id
  "name": "rtmp1", //名称 --非空
  "type": "rtmp", //类型
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "url": "rtmp://192.168.0.73:1935/live/vv", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "user": "", //鉴权账户
```

```
    "password":"" //鉴权密码
}

//修改 rtmps
{
    "stream_id": "d054f1d80ed71583392fada325a02823", //stream_id
    "name": "rtmps", //名称 --非空
    "type": "rtmps", //类型
    "url": "rtmps://192.168.0.73:1935/live/vw", //拉流地址
    "connect_speed":5000, //连接速度
    "audio_sync_compst":0, //音频同步补偿
    "buffer": "live:0:0:0:0", //缓存
    "user":""," //鉴权账户
    "password":"" //鉴权密码
}

//修改 udp
{
    "stream_id": "05b8ba7a42f01af957d2e8483857962f", //stream_id
    "name": "udp", //名称 --非空
    "type": "udp", //类型 --非空
    "connect_speed":5000, //连接速度
    "audio_sync_compst":0, //音频同步补偿
    "url": "udp://192.168.36.222:554/ch01", //拉流地址
    "buffer": "live:0:0:0:0" //缓存
}

//修改 http
{
    "stream_id": "05b8ba7a42f01af957d2e8483857962f", //stream_id
    "name": "http", //名称 --非空
    "type": "http", //类型 --非空
    "connect_speed":5000, //连接速度
    "audio_sync_compst":0, //音频同步补偿
    "url": "http://192.168.36.222:554/ch01", //拉流地址
    "buffer": "live:0:0:0:0" //缓存
```

```
}  
  
//修改 https  
{  
  "stream_id": "05b8ba7a42f01af957d2e8483857962f", //stream_id  
  "name": "https", //名称 --非空  
  "type": "https", //类型 --非空  
  "url": "https://192.168.36.222:554/ch01", //拉流地址  
  "buffer": "live:0:0:0:0", //缓存  
  "connect_speed":5000, //连接速度  
  "audio_sync_compst":0, //音频同步补偿  
  "user": "", //鉴权账号 --可为空  
  "password": "" //鉴权密码--可为空  
}  
  
//修改 rtsp  
{  
  "stream_id": "05b8ba7a42f01af957d2e8483857962f", //stream_id  
  "name": "rtsp1", //名称 --非空  
  "type": "rtsp", //类型  
  "url": "rtsp://192.168.36.222:554/ch01", //拉流地址  
  "buffer": "live:0:0:0:0", //缓存  
  "trans_mode": "tcp", //rtsp 传输模式 tcp or udp  
  "connect_speed":5000, //连接速度  
  "audio_sync_compst":0, //音频同步补偿  
  "user": "", //鉴权账号 --可为空  
  "password": "" //鉴权密码--可为空  
}  
  
//修改 rtp  
{  
  "stream_id": "8ef8bd5971763f671d81efa994549764", //stream_id  
  "name": "rtp1", //名称  
  "type": "rtp", //类型  
  "connect_speed":5000, //连接速度  
  "audio_sync_compst":0, //音频同步补偿
```

```
"buffer": "live:0:0:0:0", //缓存
"sdp_content": "SDP:\nv=0\no=- 0 0 IN IP4 127.0.0.1\ns=No Name\nc=IN IP4 192.168.36.197\nt=0
0\na=tool:libavformat 61.1.100\nm=audio 1236 RTP/AVP 97\nb=AS:768\na=rtpmap:97 PCMU/48000/2" //sdp
内容
}

//修改 SRT(Listener)
{
  "stream_id": "8eca38e84020aba241ead45094c7dd24", //stream_id
  "name": "srt_listener", //名称
  "type": "srt", //类型
  "url": "srt://:1025", //srt listener 模式没有拉流 url. 固定传: srt://:port
  "buffer": "live:0:0:0:0", //缓存
  "connection_mode": "Listener", //srt 连接模式 Rendezvous or Listener or Caller
  "listener_port": 1025, //srt 连接端口
  "latency": 125, //IP 生存时间
  "show_advanced": "1", //是否显示高级设置
  "srt_id": "111", //srt id
  "encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
  "passphrase": "", //加密密码
  "bandwidth": 25, //带宽开销
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "payload_size": 1316 //载荷大小
}

//修改 SRT(Caller)
{
  "stream_id": "8eca38e84020aba241ead45094c7dd24", //stream_id
  "name": "srt_caller", //名称
  "type": "srt", //类型
  "url": "srt://192.168.36.59:1025", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "connection_mode": "Caller", //srt 连接模式 Rendezvous or Listener or Caller
  "address": "192.168.36.51", //公网 IP,需要跟 url 中的保持一致
  "listener_port": 1025, //srt 连接端口
```

```
"latency": 125, //IP 生存时间
"show_advanced": "1", //是否显示高级
"srt_id": "111", //srt id
"encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
"passphrase": "", //加密密码
"connect_speed": 5000, //连接速度
"audio_sync_compst": 0, //音频同步补偿
"bandwidth": 25, //带宽开销
"payload_size": 1316 //载荷大小
}
```

```
//修改 SRT(Rendezvous)
```

```
{
  "stream_id": "52005872cd91b13767660d196c463879", //stream_id
  "name": "srt3", //名称
  "type": "srt", //类型
  "url": "srt://192.168.36.51:1025", //拉流地址
  "buffer": "live:0:0:0:0", //缓存
  "connect_speed": 2, // 连接相应速度--确认取消，接口中暂时保留，可固定传 0
  "audio_sync_compst": 0, //音频同步补偿 --确认取消，接口中暂时保留，可固定传 0
  "connection_mode": "Rendezvous", //srt 连接模式 Rendezvous or Listener or Caller
  "address": "192.168.36.51", //公网 IP
  "listener_port": 1025, ///srt 连接端口
  "latency": 125, //IP 生存时间
  "show_advanced": "1", //是否显示高级设置
  "connect_speed": 5000, //连接速度
  "audio_sync_compst": 0, //音频同步补偿
  "srt_id": "111", //srt id
  "encryption": "none", //加密密码。用于生成加密密钥 none 不加密 AES-128/AES-192/AES-256
  "passphrase": "", //加密密码
  "bandwidth": 25, //带宽开销
  "payload_size": 1316 //载荷大小
}
```

```
//修改 rtmp_server
```

```
{
```

```

"stream_id": "8eca38e84020aba241ead45094c7dd24", //stream_id
"name": "rtmp server1", //名称
"type": "rtmp_server", //类型
"url": "rtmp://192.168.36.59:1935/live/xxx/11111", //拉流地址
"buffer": "live:0:0:0:0", //缓存
"application": "live", //rtmp 应用名称
"session": "11111", //会话名称
"security": "token", //验证方式 "" or "token"
"token": "xxx", //token 内容
"connect_speed": 5000, //连接速度
"audio_sync_compst": 0, //音频同步补偿
"enable": 1, //默认 1
"rtmp_server_port": 1935 //rtmp 端口 大于 1025 小于 65535
}

//修改 NDI
{
"stream_id": "c0343e3a2fed40071b1b51a4af53f3c3", //流 id
"type": "ndi", //类型
"url": "192.168.36.155:5963", //url
"buffer": "live:1000:1000:1500:0:0", //缓存
"trans_mode": "multicast", //NDI 连接模式 tcp/multicast/unicast/rudp
"ndi_name": "N6-21-1080P60 (Channel-NDI)", //NDI 名称
"connect_speed": 5000, //连接速度
"audio_sync_compst": 0, //音频同步补偿
"name": "N6-21-1080P60", //名称
"preview_mode": "Full" //NDI 拉流模式 Full or Preview or Auto default Auto
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» group_id	body	string	是		所属组 ID
» stream_id	body	string	是		流 id
» name	body	string	是		名称

» type	body	string	是		流 类 型 支 持 rtmp/rtmps/rtsp/udp/http/https/srt/rtp/ndi
» url	body	string	是		流 url
» buffer	body	string	是		缓存时间
» address	body	string	是		公网 ip
» application	body	string	是		rtmp_server 应用名称
» audio_sync_compst	body	integer	是		保留字段
» bandwidth	body	integer	是		srt 恢复带宽开销
» connect_speed	body	integer	是		保留字段
» connection_mode	body	string	是		srt 连接模式 Rendezvous or Listener or Caller
» enable	body	integer	是		保留字段
» encryption	body	string	是		srt 加密方式
» id	body	string	是		流 id,唯一值
» latency	body	integer	是		Srt 数据包延迟
» listener_port	body	integer	是		端口
» ndi_name	body	string	是		NDI 名称
» passphrase	body	string	是		srt 字符密钥
» password	body	string	是		连接鉴权密码
» payload_size	body	integer	是		srt 加载大小
» preview_mode	body	string	是		加载模式 Auto (根据输出分辨率自动选择主码流或子码流) Full(主码流) Preview(子码流)
» rtmp_server_port	body	integer	是		rtmp_server 端口
» sdp_content	body	string	是		rtp 协议 sdp 内容
» security	body	string	是		rtmp_server 验证方式 none or token
» session	body	string	是		rtmp_server 会话
» show_advanced	body	string	是		是否显示高级设置
» srt_id	body	string	是		srt 协议的流 id
» status	body	integer	是		拉流状态 0 未连接 1 正在连接 2 连接成功 3 连接失败 4 性能受限制, 停止拉流
» token	body	string	是		rtmp_server 鉴权 token
» trans_mode	body	string	是		rtsp 连接模式 udp/tcp
» user	body	string	是		连接鉴权账号
» connect_speed	integer	true	none		连接超时时间

» audio_synccompst	integer	true	none		音频同步补偿(ms)
--------------------	---------	------	------	--	------------

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(七) POST /source/groups/streams/remove

POST /source/groups/streams/remove

删除某一视频流

Body 请求参数

```
{
  "group_id": 1,
  "stream_id": 1
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» group_id	body	string	是		组 id
» stream_id	body	string	是		流 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(八) POST /source/groups/streams/copy

POST /source/groups/streams/copy

信号源的复制

Body 请求参数

```
{
  "from_group_id": "83b50ca8e8af2ec13f35b251b50076dd",
  "from_stream_id": "17dd78a37db89a20eb4a11a7705dafb2",
  "to_group_id": "354ee7bf5984462d483151fcc98f74fb"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» from_group_id	body	string	是		转出 group
» from_stream_id	body	string	是		转 出 stream_id
» to_group_id	body	string	是		转入 group

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(九) POST /source/groups/streams/move

POST /source/groups/streams/move

Body 请求参数

```
{
  "from_group_id": "83b50ca8e8af2ec13f35b251b50076dd",
  "from_stream_id": "17dd78a37db89a20eb4a11a7705dafb2",
  "to_group_id": "354ee7bf5984462d483151fcc98f74fb"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» from_group_id	body	string	是		none
» from_stream_id	body	string	是		none
» to_group_id	body	string	是		none

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

六、LAYOUT 模块

(一) GET /layout/template/list

GET /layout/template/list

获取布局模板列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{  
  "count": 1000,  
  "data": [  
    {  
      "name": "Single",  
      "position": [  

```

```
{
  "height": 100,
  "id": 1,
  "number": 1,
  "stream_id": "",
  "width": 100,
  "x": 0,
  "y": 0,
  "z_index": 0
}
```

```
],
"template_count": 1,
"template_id": 1
```

```
},
```

```
{
  "name": "PIP",
  "position": [
    {
      "height": 100,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 100,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
```

```
{
  "height": 30,
  "id": 2,
  "number": 2,
  "stream_id": "",
  "width": 30,
  "x": 69,
  "y": 69,
  "z_index": 1
}
```

```
    }
  ],
  "template_count": 2,
  "template_id": 2
},
{
  "name": "2x2",
  "position": [
    {
      "height": 50,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 50,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
    {
      "height": 50,
      "id": 2,
      "number": 2,
      "stream_id": "",
      "width": 50,
      "x": 50,
      "y": 0,
      "z_index": 1
    },
    {
      "height": 50,
      "id": 3,
      "number": 3,
      "stream_id": "",
      "width": 50,
      "x": 0,
      "y": 50,
```

```
    "z_index": 2
  },
  {
    "height": 50,
    "id": 4,
    "number": 4,
    "stream_id": "",
    "width": 50,
    "x": 50,
    "y": 50,
    "z_index": 3
  }
],
"template_count": 4,
"template_id": 3
},
{
  "name": "3x3",
  "position": [
    {
      "height": 33.33,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 33.33,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
    {
      "height": 33.33,
      "id": 2,
      "number": 2,
      "stream_id": "",
      "width": 33.33,
      "x": 33.33,
```

```
"y": 0,  
"z_index": 1  
},  
{  
"height": 33.33,  
"id": 3,  
"number": 3,  
"stream_id": "",  
"width": 33.33,  
"x": 66.67,
```

```
"y": 0,  
"z_index": 2  
},  
{  
"height": 33.33,  
"id": 4,  
"number": 4,  
"stream_id": "",  
"width": 33.33,  
"x": 0,  
"y": 33.33,  
"z_index": 3
```

```
},  
{  
"height": 33.33,  
"id": 5,  
"number": 5,  
"stream_id": "",  
"width": 33.33,  
"x": 33.33,  
"y": 33.33,  
"z_index": 4
```

```
},  
{  
"height": 33.33,  
"id": 6,
```

```
"number": 6,  
"stream_id": "",  
"width": 33.33,  
"x": 66.67,  
"y": 33.33,  
"z_index": 5
```

```
},
```

```
{
```

```
"height": 33.33,  
"id": 7,
```

```
"number": 7,
```

```
"stream_id": "",
```

```
"width": 33.33,
```

```
"x": 0,
```

```
"y": 66.67,
```

```
"z_index": 6
```

```
},
```

```
{
```

```
"height": 33.33,
```

```
"id": 8,
```

```
"number": 8,
```

```
"stream_id": "",
```

```
"width": 33.33,
```

```
"x": 33.33,
```

```
"y": 66.67,
```

```
"z_index": 7
```

```
},
```

```
{
```

```
"height": 33.33,
```

```
"id": 9,
```

```
"number": 9,
```

```
"stream_id": "",
```

```
"width": 33.33,
```

```
"x": 66.67,
```

```
"y": 66.67,
```

```
"z_index": 8
```

```
    }  
  ],  
  "template_count": 9,  
  "template_id": 4  
},  
{  
  "name": "4x4",  
  "position": [  
    {  
      "height": 25,  
      "id": 1,  
      "number": 1,  
      "stream_id": "",  
      "width": 25,  
      "x": 0,  
      "y": 0,  
      "z_index": 0  
    },  
    {  
      "height": 25,  
      "id": 2,  
      "number": 2,  
      "stream_id": "",  
      "width": 25,  
      "x": 25,  
      "y": 0,  
      "z_index": 1  
    },  
    {  
      "height": 25,  
      "id": 3,  
      "number": 3,  
      "stream_id": "",  
      "width": 25,  
      "x": 50,  
      "y": 0,
```

```
"z_index": 2
},
{
  "height": 25,
  "id": 4,
  "number": 4,
  "stream_id": "",
  "width": 25,
  "x": 75,
  "y": 0,
```

```
"z_index": 3
},
{
  "height": 25,
  "id": 5,
  "number": 5,
  "stream_id": "",
  "width": 25,
  "x": 0,
  "y": 25,
  "z_index": 4
```

```
},
{
  "height": 25,
  "id": 6,
  "number": 6,
  "stream_id": "",
  "width": 25,
  "x": 25,
  "y": 25,
  "z_index": 5
```

```
},
{
  "height": 25,
  "id": 7,
  "number": 7,
```

```
"stream_id": "",
"width": 25,
"x": 50,
"y": 25,
"z_index": 6
},
{
"height": 25,
"id": 8,
"number": 8,
"stream_id": "",
"width": 25,
"x": 75,
"y": 25,
"z_index": 7
},
{
"height": 25,
"id": 9,
"number": 9,
"stream_id": "",
"width": 25,
"x": 0,
"y": 50,
"z_index": 8
},
{
"height": 25,
"id": 10,
"number": 10,
"stream_id": "",
"width": 25,
"x": 25,
"y": 50,
"z_index": 9
},
}
```

```
{  
  "height": 25,  
  "id": 11,  
  "number": 11,  
  "stream_id": "",  
  "width": 25,  
  "x": 50,  
  "y": 50,  
  "z_index": 10  
},
```

```
{  
  "height": 25,  
  "id": 12,  
  "number": 12,  
  "stream_id": "",  
  "width": 25,  
  "x": 75,  
  "y": 50,  
  "z_index": 11  
},
```

```
{  
  "height": 25,  
  "id": 13,  
  "number": 13,  
  "stream_id": "",  
  "width": 25,  
  "x": 0,  
  "y": 75,  
  "z_index": 12  
},
```

```
{  
  "height": 25,  
  "id": 14,  
  "number": 14,  
  "stream_id": "",  
  "width": 25,
```

```

    "x": 25,
    "y": 75,
    "z_index": 13
  },
  {
    "height": 25,
    "id": 15,
    "number": 15,
    "stream_id": "",
    "width": 25,
    "x": 50,
    "y": 75,
    "z_index": 14
  },
  {
    "height": 25,
    "id": 16,
    "number": 16,
    "stream_id": "",
    "width": 25,
    "x": 75,
    "y": 75,
    "z_index": 15
  }
],
"template_count": 16,
"template_id": 5
}
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» count	integer	true	none		none

» data	\[object]	true	none		none
»» name	string	true	none		名称
»» position	\[object]	true	none		位置
»»» height	integer	true	none		高度 单位:百分比
»»» id	integer	true	none		id
»»» number	integer	true	none		位于某一个
»»» stream_id	string	true	none		当前位置绑定的信号源
»»» width	integer	true	none		宽度 单位:百分比
»»» x	integer	true	none		x轴位置 单位:百分比
»»» y	integer	true	none		y轴位置 单位:百分比
»»» z_index	integer	true	none		z轴
»» template_count	integer	true	none		模版数量
»» template_id	integer	true	none		模板 id
» result	string	true	none		none

(二) GET /layout/list

GET /layout/list

获取当前布局列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": [
    {
      "layout_count": 1, //布局窗格数量
      "layout_id": 1, //布局 id
      "name": "Single", //布局名称
    }
  ]
}
```

```
"position": [ //窗格位置
  {
    "height": 100, //窗格高度 单位：百分比
    "id": 1, //窗格 id
    "number": 1, //窗格下标
    "stream_id": "", //窗格当前流 id
    "width": 100, //窗格宽度 单位：百分比
    "x": 0, //窗格 x 轴 单位：百分比
    "y": 0, //窗格 y 轴 单位：百分比
    "z_index": 0 //窗格 z 轴
  }
],
"used": false
},
{
  "layout_count": 2,
  "layout_id": 2,
  "name": "PIP",
  "position": [
    {
      "height": 100,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 100,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
    {
      "height": 30,
      "id": 2,
      "number": 2,
      "stream_id": "",
      "width": 30,
      "x": 69,
```

```
        "y": 69,  
        "z_index": 1  
    }  
],  
"used": false  
},  
{  
    "layout_count": 4,  
    "layout_id": 3,  
    "name": "2x2",  
    "position": [  
        {  
            "height": 50,  
            "id": 1,  
            "number": 1,  
            "stream_id": "",  
            "width": 50,  
            "x": 0,  
            "y": 0,  
            "z_index": 0  
        },  
        {  
            "height": 50,  
            "id": 2,  
            "number": 2,  
            "stream_id": "",  
            "width": 50,  
            "x": 50,  
            "y": 0,  
            "z_index": 1  
        },  
        {  
            "height": 50,  
            "id": 3,  
            "number": 3,  
            "stream_id": "",
```

```
        "width": 50,
        "x": 0,
        "y": 50,
        "z_index": 2
    },
    {
        "height": 50,
        "id": 4,
        "number": 4,
        "stream_id": "",
        "width": 50,
        "x": 50,
        "y": 50,
        "z_index": 3
    }
],
"used": false
},
{
    "layout_count": 9,
    "layout_id": 4,
    "name": "3x3",
    "position": [
        {
            "height": 33.33,
            "id": 1,
            "number": 1,
            "stream_id": "",
            "width": 33.33,
            "x": 0,
            "y": 0,
            "z_index": 0
        },
        {
            "height": 33.33,
            "id": 2,
```

```
    "number": 2,  
    "stream_id": "",  
    "width": 33.33,  
    "x": 33.33,  
    "y": 0,  
    "z_index": 1  
  },  
  {
```

```
    "height": 33.33,  
    "id": 3,
```

```
    "number": 3,  
    "stream_id": "",  
    "width": 33.33,  
    "x": 66.67,  
    "y": 0,  
    "z_index": 2  
  },
```

```
  {  
    "height": 33.33,  
    "id": 4,  
    "number": 4,  
    "stream_id": "",  
    "width": 33.33,  
    "x": 0,  
    "y": 33.33,  
    "z_index": 3  
  },
```

```
  {  
    "height": 33.33,  
    "id": 5,  
    "number": 5,  
    "stream_id": "",  
    "width": 33.33,  
    "x": 33.33,  
    "y": 33.33,  
    "z_index": 4  
  }
```

```
},  
{  
  "height": 33.33,  
  "id": 6,  
  "number": 6,  
  "stream_id": "",  
  "width": 33.33,  
  "x": 66.67,  
  "y": 33.33,  
  "z_index": 5
```

```
},  
{  
  "height": 33.33,  
  "id": 7,  
  "number": 7,  
  "stream_id": "",  
  "width": 33.33,  
  "x": 0,  
  "y": 66.67,  
  "z_index": 6
```

```
},  
{  
  "height": 33.33,  
  "id": 8,  
  "number": 8,  
  "stream_id": "",  
  "width": 33.33,  
  "x": 33.33,  
  "y": 66.67,  
  "z_index": 7
```

```
},  
{  
  "height": 33.33,  
  "id": 9,  
  "number": 9,  
  "stream_id": "",
```

```
        "width": 33.33,  
        "x": 66.67,  
        "y": 66.67,  
        "z_index": 8  
    }  
],  
"used": false  
},  
{
```

```
    "layout_count": 16,  
    "layout_id": 5,  
    "name": "4x4",  
    "position": [  
        {  
            "height": 25,  
            "id": 1,  
            "number": 1,  
            "stream_id": "",  
            "width": 25,  
            "x": 0,  
            "y": 0,  
            "z_index": 0  
        },  
        {  
            "height": 25,  
            "id": 2,  
            "number": 2,  
            "stream_id": "",  
            "width": 25,  
            "x": 25,  
            "y": 0,  
            "z_index": 1  
        },  
        {  
            "height": 25,  
            "id": 3,
```

```
    "number": 3,  
    "stream_id": "",  
    "width": 25,  
    "x": 50,  
    "y": 0,  
    "z_index": 2  
  },  
  {
```

```
    "height": 25,  
    "id": 4,
```

```
    "number": 4,  
    "stream_id": "",  
    "width": 25,  
    "x": 75,  
    "y": 0,  
    "z_index": 3  
  },  
  {
```

```
    "height": 25,  
    "id": 5,  
    "number": 5,  
    "stream_id": "",  
    "width": 25,  
    "x": 0,  
    "y": 25,  
    "z_index": 4
```

```
  },  
  {  
    "height": 25,  
    "id": 6,  
    "number": 6,  
    "stream_id": "",  
    "width": 25,  
    "x": 25,  
    "y": 25,  
    "z_index": 5
```

```
},  
{  
  "height": 25,  
  "id": 7,  
  "number": 7,  
  "stream_id": "",  
  "width": 25,  
  "x": 50,  
  "y": 25,  
  "z_index": 6
```

```
},  
{  
  "height": 25,  
  "id": 8,  
  "number": 8,  
  "stream_id": "",  
  "width": 25,  
  "x": 75,  
  "y": 25,  
  "z_index": 7
```

```
},  
{  
  "height": 25,  
  "id": 9,  
  "number": 9,  
  "stream_id": "",  
  "width": 25,  
  "x": 0,  
  "y": 50,  
  "z_index": 8
```

```
},  
{  
  "height": 25,  
  "id": 10,  
  "number": 10,  
  "stream_id": "",
```

```
    "width": 25,  
    "x": 25,  
    "y": 50,  
    "z_index": 9  
  },  
  {  
    "height": 25,  
    "id": 11,  
    "number": 11,  
    "stream_id": "",
```

```
    "width": 25,  
    "x": 50,  
    "y": 50,  
    "z_index": 10  
  },  
  {  
    "height": 25,  
    "id": 12,  
    "number": 12,  
    "stream_id": "",  
    "width": 25,  
    "x": 75,  
    "y": 50,  
    "z_index": 11
```

```
  },  
  {  
    "height": 25,  
    "id": 13,  
    "number": 13,  
    "stream_id": "",  
    "width": 25,  
    "x": 0,  
    "y": 75,  
    "z_index": 12
```

```
  },  
  {
```

```
        "height": 25,
        "id": 14,
        "number": 14,
        "stream_id": "",
        "width": 25,
        "x": 25,
        "y": 75,
        "z_index": 13
    },
    {
        "height": 25,
        "id": 15,
        "number": 15,
        "stream_id": "",
        "width": 25,
        "x": 50,
        "y": 75,
        "z_index": 14
    },
    {
        "height": 25,
        "id": 16,
        "number": 16,
        "stream_id": "",
        "width": 25,
        "x": 75,
        "y": 75,
        "z_index": 15
    }
],
"used": false
}
],
"result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» layout_count	integer	true	none		布局窗格数量
»» layout_id	integer	true	none		布局 id
»» name	string	true	none		布局名称
»» position	\[object]	true	none		窗格位置
»»» height	integer	true	none		窗格高度 单位: 百分比
»»» id	integer	true	none		窗格 id
»»» number	integer	true	none		窗格下标
»»» stream_id	string	true	none		窗格当前流 id
»»» width	integer	true	none		窗格宽度 单位: 百分比
»»» x	integer	true	none		窗格 x 轴 单位: 百分比
»»» y	integer	true	none		窗格 y 轴 单位: 百分比
»»» z_index	integer	true	none		窗格 z 轴
»» used	boolean	true	none		none
» result	string	true	none		none

(三) POST /layout/add

POST /layout/add

添加布局模板

Body 请求参数

```
{
  "template_id": 1, //模版 id
  "name": "AddSingle2" //布局名称
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
----	----	----	----	-----	----

app	header	string	是		none
body	body	object	否		none
» template_id	body	integer	是		模版 id
» name	body	string	是		布局名称

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(四) POST /layout/remove

POST /layout/remove

删除预设

Body 请求参数

```
{
  "layout_id": 7 //布局 id
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» layout_id	body	integer	是		布局 id

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(五) POST /layout/select

POST /layout/select

给某个 output 选择布局

Body 请求参数

```
{
  "output_id": "1", //输出 ID
  "layout_id": 2 //布局 ID
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID
» layout_id	body	integer	是		布局 ID

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /layout/modify

POST /layout/modify

修改某一预设

Body 请求参数

```
{
  "layout_id": 6, //布局 id
  "position": [
    {
      "width": 30, //宽度 单位:百分比
      "height": 30, //高度 单位:百分比
      "number": 2, //下标
      "z_index": 1, //z 轴
      "id": 2, //位置 ID
      "x": 0, //x 轴位置 单位: 百分比
      "y": 0 //y 轴位置 单位: 百分比
    }
  ]
}
```

```

}
],
"output_id": "1" //输出 id
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» layout_id	body	integer	是		布局 id
» position	body	\[object]	是		none
»» width	body	integer	否		宽度 单位:百分比
»» height	body	integer	否		高度 单位:百分比
»» number	body	integer	否		下标
»» z_index	body	integer	否		z 轴
»» id	body	integer	否		位置 ID
»» x	body	integer	否		x轴位置 单位:百分比
»» y	body	integer	否		y轴位置 单位:百分比
» output_id	body	string	是		输出 id

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(七) POST /layout/save

POST /layout/save

保存 outpu 中的修改到某一布局

Body 请求参数

```

{
"output_id": "1", //输出 ID, 支持 1, 2 两个布局

```

```
"layout_id": 6 // 布局 id
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID, 支持 1, 2 两个布局
» layout_id	body	integer	是		布局 id

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(八) POST /layout/resave

POST /layout/resave

预设另存为

Body 请求参数

```
{
  "output_id": "1", //输出 ID
  "layout_id": 6, //布局 ID
  "layout_name": "Test" //布局名称
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID
» layout_id	body	integer	是		布局 ID
» layout_name	body	string	是		布局名称

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(九) POST /layout/reload

POST /layout/reload

重新加载 output 中的布局

Body 请求参数

```
{
  "output_id": 1,
  "layout_name": "5",
  "layout_id": 6
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	integer	是		none
» layout_id	body	integer	是		none

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十) POST /layout/rename

POST /layout/rename

修改某一布局名称

Body 请求参数

```
{
  "layout_id": 1,
  "name": ""
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» layout_id	body	integer	是		布局 id
» name	body	string	是		名称

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

七、OUTPUT 模块

(一) POST /output/interfaces/get

POST /output/interfaces/get

获取输出接口类型列表

Body 请求参数

```
{
  "output_id": 1,
  "type": "audio"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否		none
type	query	string	否		输出类型： video, audio
app	header	string	是		none
body	body	object	否		none

» output_id	body	string	是	布局 id	none
» type	body	string	是	获取接口类型	支持 video/audio 分别获取视频 输出接口信息 以及音频输出 接口信息

返回示例

```
{
  "data": [
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 1,
      "label": "HDMI1",
      "mode": "HDMI",
      "support": false,
      "visible": true
    },
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 2,
      "label": "HDMI2",
      "mode": "HDMI",
      "support": true,
      "visible": true
    },
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 3,
      "label": "SDI",
      "mode": "HDMI",
      "support": true,
      "visible": true
    }
  ]
}
```

```
    }  
  ],  
  "result": "ok"  
}  
{  
  "data": [  
    {  
      "id": 1,  
      "label": "HDMI",  
      "enable": false,  
      "samplerate": 44.1,  
      "channels": 2,  
      "volume": 0,  
      "mute": false,  
      "visible": true  
    },  
    {  
      "id": 2,  
      "label": "HDMI1",  
      "enable": false,  
      "samplerate": 44.1,  
      "channels": 2,  
      "volume": 0,  
      "mute": false,  
      "visible": true  
    },  
    {  
      "id": 3,  
      "label": "SDI",  
      "enable": false,  
      "samplerate": 44.1,  
      "channels": 2,  
      "volume": 0,  
      "mute": false,  
      "visible": true  
    }  
  ],  
}
```

```

{
  "id": 4,
  "label": "LineOut",
  "enable": false,
  "samplerate": 44.1,
  "channels": 2,
  "volume": 0,
  "mute": false,
  "visible": true
}
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» colorspace	string	true	none		当前输出接口色彩模式
»» enable	boolean	true	none		是否开启
»» id	integer	true	none		输出接口 id
»» label	string	true	none		输出接口显示名称
»» mode	string	true	none		输出接口类型
»» support	boolean	true	none		当前设备是否支持当前接口
»» visible	boolean	true	none		输出接口是否开启
» result	string	true	none		none

(二) POST /output/interfaces/set(audio)

POST /output/interfaces/set

Body 请求参数

```

{
  "channels": 2, //通道数

```

```

"enable": true, //是否开启
"id": 1, //id
"label": "HDMI1", //显示名称
"mute": true, //是否静音
"samplerate": 48000, //采样数
"visible": true, //是否支持
"volume": 14, //声音增大 dB
"output_id": "1", //输出 id
"type": "audio", //类型 video or audio
"intf_id": 1 //id
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» channels	body	integer	是		通道数
» enable	body	boolean	是		是否开启
» id	body	integer	是		id
» label	body	string	是		显示名称
» mute	body	boolean	是		是否静音
» samplerate	body	integer	是		采样数
» visible	body	boolean	是		是否支持
» volume	body	integer	是		声音增大 dB
» output_id	body	string	是		输出 id
» type	body	string	是		类型 video or audio
» intf_id	body	integer	是		id

返回示例

```

{
  "result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
----	----	----	----	-----	----

» result	string	true	none		none
----------	--------	------	------	--	------

(三) GET /output/list

GET /output/list

获取窗口列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": [
    {
      "background": "#000000",
      "bg_enable": false,
      "border": {
        "bottom_width": 2,
        "color": "#FF0000",
        "enable": false,
        "hex_color": 16711680,
        "left_width": 2,
        "right_width": 2,
        "top_width": 2
      },
      "id": "1",
      "layout_id": 2,
      "modified": false,
      "name": "output1",
      "res_id": "5",
      "res_name": "1920x1080P60",
      "vumeter": true
    },
    {
      "background": "#000000",
      "bg_enable": false,
      "border": {
```

```

    "bottom_width": 2,
    "color": "#FF0000",
    "enable": false,
    "hex_color": 16711680,
    "left_width": 2,
    "right_width": 2,
    "top_width": 2
  },
  "id": "2",
  "layout_id": 1,
  "modified": false,
  "name": "output2",
  "res_id": "5",
  "res_name": "1920x1080P60",
  "vumeter": true
}
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» background	string	true	none		背景颜色
»» bg_enable	boolean	true	none		是否开启背景
»» border	object	true	none		边框参数
»»»	integer	true	none		边框宽度
bottom_width					
»»» color	string	true	none		边框颜色
»»» enable	boolean	true	none		是否开启边框
»»» hex_color	integer	true	none		none
»»» left_width	integer	true	none		边框宽度
»»»	integer	true	none		边框宽度
right_width					
»»» top_width	integer	true	none		边框宽度

»» id	string	true	none		布局 id
»» layout_id	integer	true	none		布局 id
»» modified	boolean	true	none		当前布局是否已被修改
»» name	string	true	none		布局名称
»» res_id	string	true	none		输出分辨率 id
»» res_name	string	true	none		输出分辨率名称
»» vumeter	boolean	true	none		none
» result	string	true	none		none

(四) GET /output/get

GET /output/get

获取某一窗口信息

1. 请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否		布局 ID
app	header	string	是		none

返回示例

```
{
  "data": {
    "layout_id": 2,
    "name": "output1",
    "position": [
      {
        "enable": true,
        "height": 30,
        "id": 1,
        "mute": false,
        "number": 1,
        "resolution": "1920x1080p60",
        "status": 2,
        "stream_id": "e893d03a94326b981fd3aa276f90183a",
        "stream_ip": "192.168.36.238:5965",
        "stream_name": "N60_238 (HB)",

```

```
"type": "ndi",
"video_bitrate": "55479.4",
"video_format": "NDI | HB",
"volume": 0,
"vu_meter": true,
"width": 30,
"x": 0,
"y": 0,
"z_index": 0
},
```

```
{
  "enable": false,
  "height": 30,
  "id": 2,
  "mute": false,
  "number": 2,
  "resolution": "0x0p0",
  "status": 2,
  "stream_id": "",
  "stream_ip": "",
  "stream_name": "",
  "type": "",
  "video_bitrate": "0",
  "video_format": "",
  "volume": 0,
  "vu_meter": true,
  "width": 30,
  "x": 69,
  "y": 69,
  "z_index": 1
}
],
"vumeter": true
},
"result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» layout_id	integer	true	none		布局 id
»» name	string	true	none		输出名称
»» position	\[object]	true	none		none
»»» enable	boolean	true	none		是否开启拉流
»»» height	integer	true	none		高度 单位:百分比
»»» id	integer	true	none		位置 ID
»»» mute	boolean	true	none		是否静音
»»» number	integer	true	none		位置所处下标
»»» resolution	string	true	none		输出分辨率
»»» status	integer	true	none		流状态, 参考流获取定义
»»» stream_id	string	true	none		当前位置正在加载的流 ID
»»» stream_ip	string	true	none		当前位置正在加载的流 IP
»»» stream_name	string	true	none		当前位置正在加载的流名称
»»» type	string	true	none		当前位置正在加载的流类型
»»» video_bitrate	string	true	none		当前位置正在加载的流码流
»»» video_format	string	true	none		//当前位置正在加载的流编码类型
»»» volume	integer	true	none		none
»»» vu_meter	boolean	true	none		none
»»» width	integer	true	none		宽度 单位:百分比
»»» x	integer	true	none		x轴位置 单位:百分比

»»» y	integer	true	none		y轴位置 单位： 百分比
»»» z_index	integer	true	none		none
»» vumeter	boolean	true	none		none
» result	string	true	none		none

(五) POST /output/source/set

POST /output/source/set

设置视频源到某一窗口的某个位置上

Body 请求参数

```
{
  "from": {
    "output_id": "1",
    "pos_id": 1
  },
  "to": {
    "output_id": "2",
    "pos_id": 2,
    "stream_id": ""
  }
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» from	body	object	是		从某个位置移动流
»» type	body	string	是		类型
»» stream_id	body	string	是		流 id
»» stream_name	body	string	是		流名称
»» stream_url	body	string	是		流 url
»» pos_id	body	string	是		位置 id
»» output_id	body	string	是		输出 id
»» layout_id	body	string	是		布局 id

» to	body	object	是		填充位置信息
»» type	body	string	是		类型
»» stream_id	body	string	是		流 id
»» stream_name	body	string	是		流名称
»» stream_url	body	string	是		流 url
»» pos_id	body	integer	是		位置 id
»» output_id	body	string	是		输出 id
»» layout_id	body	integer	是		布局 id

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /output/source/remove

POST /output/source/remove

删除窗口上某个位置上绑定的源信息

Body 请求参数

```
{
  "output_id": "1",
  "pos_id": 1
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是	布局 ID	(1, 2)
» pos_id	body	integer	是	位置 ID	none

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
----	----	----	----	-----	----

» result	string	true	none		none
----------	--------	------	------	--	------

(七) GET /output/resolution/list

GET /output/resolution/list

获取分辨率（时序）列表

Body 请求参数

```
{}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none

返回示例

```
{
  "data": [
    {
      "id": "1",
      "name": "3840x2160P60(HDMI only)",
      "value": "3840x2160P60(HDMI only)"
    },
    {
      "id": "2",
      "name": "3840x2160P50(HDMI only)",
      "value": "3840x2160P50(HDMI only)"
    },
    {
      "id": "3",
      "name": "3840x2160P30(HDMI only)",
      "value": "3840x2160P30(HDMI only)"
    },
    {
      "id": "4",
      "name": "3840x2160P25(HDMI only)",
      "value": "3840x2160P25(HDMI only)"
    },
    {

```

```
"id": "5",
"name": "1920x1080P60",
"value": "1920x1080P60"
},
{
  "id": "6",
  "name": "1920x1080P59.94(SDI only)",
  "value": "1920x1080P59.94(SDI only)"
},
{
  "id": "7",
  "name": "1920x1080P50",
  "value": "1920x1080P50"
},
{
  "id": "8",
  "name": "1920x1080P30",
  "value": "1920x1080P30"
},
{
  "id": "9",
  "name": "1920x1080P29.97(SDI only)",
  "value": "1920x1080P29.97(SDI only)"
},
{
  "id": "10",
  "name": "1920x1080P25",
  "value": "1920x1080P25"
},
{
  "id": "11",
  "name": "1920x1080P24",
  "value": "1920x1080P24"
},
{
  "id": "12",
```

```
"name": "1920x1080P23.98(SDI only)",
"value": "1920x1080P23.98(SDI only)"
},
{
  "id": "13",
  "name": "1920x1080i60",
  "value": "1920x1080i60"
},
{
  "id": "14",
  "name": "1920x1080i50",
  "value": "1920x1080i50"
},
{
  "id": "15",
  "name": "1920x1080i59.94(SDI only)",
  "value": "1920x1080i59.94(SDI only)"
},
{
  "id": "16",
  "name": "1280x720P60",
  "value": "1280x720P60"
},
{
  "id": "17",
  "name": "1280x720P59.94(SDI only)",
  "value": "1280x720P59.94(SDI only)"
},
{
  "id": "18",
  "name": "1280x720P50",
  "value": "1280x720P50"
},
{
  "id": "19",
  "name": "576i/625i 50Hz(SDI only)",
```

```

    "value": "576i/625i 50Hz(SDI only)"
  },
  {
    "id": "20",
    "name": "480i/525i 60Hz(SDI only)",
    "value": "480i/525i 60Hz(SDI only)"
  }
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» id	string	true	none		输出分辨率 id
»» name	string	true	none		输出分辨率名称
»» value	string	true	none		输出分辨率值
» result	string	true	none		none

(八) POST /output/resolution/set

POST /output/resolution/set

设置分辨率（时序）列表

Body 请求参数

```

{
  "output_id": "1",
  "res_index": 14
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是	输出 id	none
» res_id	body	integer	是	输出分辨率 ID	none

返回示例

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none
» data	\[object]	true	none		none
»» name	string	true	none		none
»» value	string	true	none		none

(九) GET output/audiomix/get

GET /output/audiomix/get

获取某一窗口的混音音频状态列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否		布局 id
app	header	string	是		none

返回示例

```
{
  "data": {
    "output": [
      {
        "stream_id": "1",
        "name": "n60...",
        "enable": false,
        "volume": 0,
        "mute": false,
        "type": "output"
      },
      {
        "stream_id": "2",
        "name": "n50...",
        "enable": false,
        "volume": 0,
        "mute": false,
```

```

        "type": "output"
    }
],
"preview": [
    {
        "stream_id": "1",
        "name": "n50...",
        "enable": false,
        "volume": 0,
        "mute": false,
        "type": "preview"
    }
]
}
}

```

2. 返回结果

状态码	状态码含义	说明	数据模型
200	OK	none	Inline

3. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» output	\[object]	true	none	output 类型	当前 output 音频源信息 类型分为两种：output 类型根据布局切换变动,preview 则是单个 output 共有
»»» enable	boolean	false	none		是否开启
»»» mute	boolean	false	none		是否静音
»»» name	string	false	none		名称
»»» pos_id	integer	false	none		位置 id
»»»	string	false	none		流 id

stream_id					
»»» type	string	false	none		类型
»»» volume	integer	false	none		增益 dB 值
»» preview	\[object]	true	none	预览类型	当前 preview 音频源信息
»»» enable	boolean	false	none		是否开启
»»» mute	boolean	false	none		是否静音
»»» name	string	false	none		名称
»»» stream_id	string	false	none		流 id
»»» type	string	false	none		类型
»»» volume	integer	false	none		增益 dB 值
» result	string	true	none		none

(十) POST /output/audiomix/set

POST /output/audiomix/set

设置某一窗口中的混音音频状态

Body 请求参数

```
{
  "output_id": "1",
  "stream_id": "895a2714dc8e5197fb0491066be73f22", //由 audio_id 改成 stream_id
  "enable": true,
  "volume": 0,
  "mute": false,
  "type": "output"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		布局 id
» stream_id	body	string	是		流 id
» enable	body	boolean	是		是否开启
» type	body	string	是		类型 支持

					output/preview
--	--	--	--	--	----------------

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十一) POST /output/audiomix/remove

POST /output/audiomix/remove

删除手动添加音频列表中的音频

Body 请求参数

```
{
  "output_id": "1", //布局 id
  "stream_id": "e893d03a94326b981fd3aa276f90183a", //流 id
  "type": "preview" //类型 支持 output/preview
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		布局 id
» stream_id	body	string	是		流 id
» type	body	string	是		类型 支持 output/preview

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
----	----	----	----	-----	----

» result	string	true	none		none
----------	--------	------	------	--	------

(十二) POST /output/audiomix/add

POST /output/audiomix/add

添加预览中的音频到音频列表中

Body 请求参数

```
{
  "output_id": "1",
  "stream_id": "895a2714dc8e5197fb0491066be73f22"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID
» stream_id	body	string	是		流 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十三) POST /ptz/set

POST /ptz/set

设置流的 PTZ (目前只支持 NDI)

Body 请求参数

```
{
  "output_id": "1",
  "pos_id": 1,
  "type": ["ptz_zoom", "ptz_zoom_speed", "ptz_pan_tilt", "ptz_pan_tilt_speed", "ptz_store_preset",
"ptz_recall_preset", "ptz_auto_focus", "ptz_focus", "ptz_focus_speed", "ptz_white_balance_auto",
```

```

"ptz_white_balance_indoor", "ptz_white_balance_outdoor", "ptz_white_balance_oneshot",
"ptz_white_balance_manual", "ptz_exposure_auto", "ptz_exposure_manual", "ptz_is_supported"],
  "data": {
    "zoom_value": 0.5,
    "zoom_speed": -1.0,
    "pan_value": -1.0,
    "tilt_value": -1.0,
    "pan_speed": -1.0,
    "tilt_speed": -1.0,
    "preset_no": 0,
    "speed": 0.0,
    "focus_value": 0.0,
    "focus_speed": -1.0,
    "red": 0.0,
    "blue": 0.0,
    "exposure_level": 0.0
  }
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID
» pos_id	body	integer	是		位置 id
» type	body	string	是		类型
» data	body	object	是		具体值
»» zoom_value	body	number	是		0.0 (zoomed in) ... 1.0 (zoomed out)
»» zoom_speed	body	integer	是		-1.0 (zoom outwards) ... +1.0 (zoom inwards)
»» pan_value	body	integer	是		-1.0 (left) ... 0.0

					(centered) ... +1.0 (right)
»» tilt_value	body	integer	是		-1.0 (bottom) ... 0.0 (centered) ... +1.0 (top)
»» pan_speed	body	integer	是		-1.0 (moving right) ... 0.0 (stopped) ... +1.0 (moving left)
»» tilt_speed	body	integer	是		-1.0 (down) ... 0.0 (stopped) ... +1.0 (moving up)
»» preset_no	body	integer	是		0 ... 99
»» speed	body	integer	是		0.0(as slow as possible) ... 1.0(as fast as possible)
»» focus_value	body	integer	是		0.0 (focused to infinity) ... 1.0 (focused as close as possible)
»» focus_speed	body	integer	是		-1.0 (focus outwards) ... +1.0 (focus inwards)
»» red	body	integer	是		0.0(not red) ... 1.0(very red)
»» blue	body	integer	是		0.0(not blue) ... 1.0(very blue)

»» exposure_level	body	integer	是		0.0(dark) ... 1.0(light)
----------------------	------	---------	---	--	-----------------------------

● type 字段支持参数如下

类型名称	功能说明
ptz_zoom	控制 PTZ 的变焦操作，需要 data 对象中包含 zoom_value 字段，设置变焦值
ptzzoomspeed	控制 PTZ 的变焦速度，需要 data 对象中包含 zoom_speed 字段，设置变焦速度值
ptzpartilt	控制 PTZ 的水平 (pan) 和垂直 (tilt) 转动，需要 data 对象中包含 pan_value 和 tilt_value 字段，分别设置水平和垂直转动的值
ptzpartilt_speed	控制 PTZ 的水平和垂直转动速度，需要 data 对象中包含 pan_speed 和 tilt_speed 字段，分别设置水平和垂直转动速度值
ptzstorepreset	存储 PTZ 预设位置，需要 data 对象中包含 preset_no 字段，设置预设位置编号
ptzrecallpreset	调用 PTZ 预设位置，需要 data 对象中包含 preset_no 和 speed 字段，分别设置预设位置编号和调用速度
ptzautofocus	启用自动对焦功能，无需参数
ptz_focus	控制 PTZ 的对焦操作，需要 data 对象中包含 focus_value 字段，设置对焦值
ptz focusspeed	控制 PTZ 的对焦速度，需要 data 对象中包含 focus_speed 字段，设置对焦速度值
ptzwhitebalance_auto	启用自动白平衡，无需参数
ptzwhitebalance_indoor	设置室内白平衡模式，无需参数
ptzwhitebalance_outdoor	设置室外白平衡模式，无需参数
ptzwhitebalance_oneshot	执行一次性白平衡调整，无需参数
ptzwhitebalance_manual	手动设置白平衡，需要 data 对象中包含 red (红色增益) 和 blue (蓝色增益)
ptzexposureauto	启用自动曝光，无需参数
ptzexposuremanual	手动设置曝光等级，需要 data 对象中包含 exposure_level

| ptzissupported | 查询设备是否支持 PTZ 功能 (返回布尔值)，无需参数

● 速度控制类 必须重置> 示例 // 必须重置 设置 PTZ 需要分成两步: 1、设置目标类型以及目标参数 step1:{"type":

"ptzfocusspeed", "data": {"focus_speed": 0.5}, "pos_id": 1, "output_id": "1", "layout_id": 1}2、设置目标类型以及参数
归原 step2:{"type": "ptzfocusspeed", "data": {"focus_speed": 0}, "pos_id": 1, "output_id": "1", "layout_id": 1}`

- 绝对位置类接口 无需重置 > 示例 `// 直接设置即可 1、设置目标类型以及目标参数 step1:{"type": "ptzrecall/preset", "data": {"preset_no": 1, "speed": 0.5}, "pos_id": 1, "output_id": "1", "layout_id": 1}`

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十四) POST /output/audio/sound

POST /output/audio/sound

获取音频音注

Body 请求参数

```
{  
  "output_id": "1"  
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 ID

返回示例

```
{  
  "data": [  
    {  
      "dB": [  
        0,  
        0,  
        0,  
        0  
      ]  
    }  
  ]  
}
```

```

    ],
    "stream_id": "e893d03a94326b981fd3aa276f90183a"
  },
  {
    "dB": [
      0,
      0,
      0,
      0
    ],
    "stream_id": "e893d03a94326b981fd3aa276f90183a"
  }
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» dB	\[integer]	true	none		音频 dB 值
»» stream_id	string	true	none		流 id
» result	string	true	none		none

(十五) POST /output/border/set

POST /output/border/set

设置布局边框

Body 请求参数

```

{
  "output_id": "1", //输出 id
  "enable": true, //是否开启
  "color": "#FF0000" //边框颜色
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
----	----	----	----	-----	----

app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 id
» enable	body	boolean	是		是否开启
» color	body	string	是		边框颜色

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十六) POST /output/border/get

POST /output/border/get

Body 请求参数

```
{
  "output_id": 1 //输出 ID
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	integer	是		输出 ID

返回示例

```
{
  "data": {
    "bottom_width": 2,
    "color": "#FF0000",
    "enable": true,
    "hex_color": 16711680,
    "left_width": 2,
    "right_width": 2,
  }
}
```

```

    "top_width": 2
  },
  "result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» bottom_width	integer	true	none		边框宽度
»» color	string	true	none		边框颜色
»» enable	boolean	true	none		是否卡其
»» hex_color	integer	true	none		none
»» left_width	integer	true	none		边框宽度
»» right_width	integer	true	none		边框宽度
»» top_width	integer	true	none		边框宽度
» result	string	true	none		none

(十七) POST /output/background/set

POST /output/background/set

Body 请求参数

```

{
  "output_id": "1", //输出 id
  "enable": true, //是否开启
  "background": "#000000" //背景颜色
}

```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» output_id	body	string	是		输出 id
» enable	body	boolean	是		是否开启
» background	body	string	是		背景颜色

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

(十八) GET /output/background/get

GET /output/background/get

1. 请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否		none
app	header	string	是		none

返回示例

```
{
  "data": {
    "background": "",
    "enable": true
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» background	string	true	none		背景颜色
»» enable	boolean	true	none		是否开启
» result	string	true	none		none

(十九) POST /output/audio/check

POST /output/audio/check

获取音频是否可选状态

Body 请求参数

```
{
  "id": ""
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	string	是		音频设备 ID

返回示例

```
{
  "data": true,
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	boolean	true	none		是否允许打开
» result	string	true	none		none

八、NETWORK 模块

(一) GET /network/get

GET /network/get

获取网卡列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": [
    {
      "device": "eth0", //网口名称
      "dns": "", //dns
    }
  ]
}
```

```

    "dynamic": "n", //是否 dhcp 动态分配
    "enable": 1, //是否开启
    "enable2": true, //维护地址是否开启
    "gw": "192.168.36.254", //网关
    "gw2": "", //维护地址网关
    "ip": "192.168.36.6", //ip
    "ip2": "192.168.1.163", //维护地址 ip
    "mac": "18:C3:F4:25:66:66", //mac 地址
    "mask": "255.255.255.0", //子网掩码
    "mask2": "255.255.255.0", //维护地址子网掩码

    "state": "up" //是否连接状态
  },
  {
    "device": "eth1",
    "dns": "",
    "dynamic": "n",
    "enable": 1,
    "enable2": true,
    "gw": "192.168.36.254",
    "gw2": "",
    "ip": "192.168.36.16",
    "ip2": "192.168.2.161",
    "mac": "18:C3:F4:25:66:67",
    "mask": "255.255.255.0",
    "mask2": "255.255.255.0",
    "state": "down"
  }
],
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» device	string	true	none		网口名称

»» dns	string	true	none		dns
»» dynamic	string	true	none		是否 dhcp 动态分配
»» enable	integer	true	none		是否开启
»» enable2	boolean	true	none		维护地址是否开启
»» gw	string	true	none		网关
»» gw2	string	true	none		维护地址网关
»» ip	string	true	none		ip
»» ip2	string	true	none		维护地址 ip
»» mac	string	true	none		mac 地址
»» mask	string	true	none		子网掩码
»» mask2	string	true	none		维护地址子网掩码
»» state	string	true	none		是否连接状态
» result	string	true	none		none

(二) POST /network/routing/add

POST /network/routing/add

添加路由

Body 请求参数

```
{
  "device": "eth1", //网口名称
  "ip": "192.168.1.1", //路由 ip
  "netmask": "255.255.255.255", //子网掩码
  "gw": "192.168.36.6", //网关
  "mtu": 1500, //MTU
  "metric": 0 //Metric
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» device	body	string	是		网口名称

» ip	body	string	是		路由 ip
» netmask	body	string	是		子网掩码
» gw	body	string	是		网关
» mtu	body	integer	是		MTU
» metric	body	integer	是		Metric

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) POST /network/http/set

POST /network/http/set

设置 http,https 端口

Body 请求参数

```
{
  "http": true, //是否开启 http
  "httpPort": 80, //http 端口
  "https": true, //是否开启 https
  "httpsPort": 443 //https 端口
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» http	body	boolean	是		是否开启 http
» httpPort	body	integer	是		http 端口
» https	body	boolean	是		是否开启 https
» httpsPort	body	integer	是		https 端口

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(四) GET /network/http/get

GET /network/http/get

获取 http、https 端口

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": {
    "http": true,
    "httpPort": 80,
    "https": true,
    "httpsPort": 443
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» http	boolean	true	none		是否开启 http
»» httpPort	integer	true	none		http 端口
»» https	boolean	true	none		是否开启 https
»» httpsPort	integer	true	none		https 端口
» result	string	true	none		none

(五) POST /network/ping

POST /network/ping

查看是否能 ping 通指定地址

Body 请求参数

```
{
  "ip": "192.168.36.124"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» ip	body	string	是		目标地址

返回示例

```
{
  "data": {
    "lossRate": 0,
    "reachable": true
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» lossRate	integer	true	none		丢包率
»» reachable	boolean	true	none		是否可达
» result	string	true	none		none

(六) POST /network/routing/set

POST /network/routing/set

设置路由状态（是否启用）

Body 请求参数

```
{
  "id": "c26163a66109b6c1a717e7be658ab071", //唯一值
  "status": 0 //是否开启 1 为开启 0 为关闭
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	string	是		唯一值
» status	body	integer	是		是否开启 1 为开启 0 为关闭

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(七) POST /network/routing/del

POST /network/routing/del

删除路由

Body 请求参数

```
{
  "ids": "e762817bb4e7d6889bc4cd09477672a2" //id 唯一值
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» ids	body	string	是		id 唯一值

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(八) POST /network/routing/modify

POST /network/routing/modify

修改路由

Body 请求参数

```
{
  "id": "e762817bb4e7d6889bc4cd09477672a2", //id 唯一值
  "device": "eth1", //网口名称
  "ip": "192.168.1.2", //ip
  "netmask": "255.255.255.255", //子网掩码
  "gw": "192.168.36.6", //网关
  "mtu": 1500, //mtu
  "metric": 0 // metric
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	string	是		id 唯一值
» device	body	string	是		网口名称
» ip	body	string	是		ip
» netmask	body	string	是		子网掩码
» gw	body	string	是		网关
» mtu	body	integer	是		mtu
» metric	body	integer	是		none

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(九) GET network/routing/get

GET /network/routing/get

获取路由列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "result": "ok",
  "data": [
    {
      "id": 1,
      "device": "eth0",
      "ip": "192.168.38.12",
      "netmask": "255.255.255.0",
      "gw": "192.168.38.254",
      "mtu": 1500,
      "metric": 0,
      "status": 0 //是否启用
    }
  ]
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none

»» device	string	false	none		网口名称
»» gw	string	false	none		网关
»» id	string	false	none		id
»» ip	string	false	none		ip
»» metric	integer	false	none		metric
»» mtu	integer	false	none		mtu
»» netmask	string	false	none		子网掩码
»» status	integer	false	none		状态
» result	string	true	none		none

(十) POST /network/set

POST /network/set

修改网络设置

Body 请求参数

```
{
  "device": "eth0", //网口名称
  "dynamic": "n", //是否 dhcp
  "ip": "192.168.36.6", //ip
  "mask": "255.255.255.0", //子网掩码
  "gw": "192.168.36.254", //网关
  "dns": "", //dns
  "mac": "18:C3:F4:25:66:66" //mac 地址
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» device	body	string	是		网口名称
» dynamic	body	string	是		是否 dhcp
» ip	body	string	是		ip
» mask	body	string	是		子网掩码
» gw	body	string	是		网关
» dns	body	string	是		dns
» mac	body	string	是		mac 地址

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(十一) POST /network/modify

POST /network/modify

修改维护地址

Body 请求参数

```
{
  "device": "eth0", //网口名称
  "enable2": true, //是否开启
  "ip2": "192.168.1.168", //维护地址 ip
  "mask2": "255.255.255.0", //维护地址子网掩码
  "gw2": "" //维护地址子网网关
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» device	body	string	是		网口名称
» enable2	body	boolean	是		是否开启
» ip2	body	string	是		维护地址 ip
» mask2	body	string	是		维护地址子网掩码
» gw2	body	string	是		维护地址子网网关

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

九、SYSTIME 模块

(一) GET /timeregion/settings

GET /timeregion/settings

获取时区设置信息

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": {
    "ntp_servers": "ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com",
    "offset": 8,
    "time": "2024-09-09 10:40:46",
    "timetype": "pc",
    "timezone": "Asia/Shanghai"
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» ntp_servers	string	true	none		ntp 服务器地址

»» offset	integer	true	none		时区偏移
»» time	string	true	none		时间字符串
»» timetype	string	true	none		3 种时间类型： ntp,pc,manual
»» timezone	string	true	none		时区名称
» result	string	true	none		none

(二) POST /timeregion/time/set

POST /timeregion/time/set

时间设置

Body 请求参数

```
{
  "time": "2024-09-09 09:20:46",
  "timetype": "pc",
  "offset": 8,
  "ntp_servers": "ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» time	body	string	是		时间字符串形式： "2024-09-09 09:20:46"
» timetype	body	string	是		时间类型，三种情况： ntp,pc,manual
» offset	body	integer	是		时区偏移，如 8
» ntp_servers	body	string	是		ntp 服务器 Url 地址,非 ntp 时,传空或不传

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) POST /timeregion/timezone/set

POST /timeregion/timezone/set

时区设置

Body 请求参数

```
{
  "timezone": "Asia/Shanghai",
  "offset": 8
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» timezone	body	string	是		时区名称
» offset	body	integer	是		时区偏移

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

十、SYSTEM 模块

(一) GET /sys/reboot

GET /sys/reboot

系统重启

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(二) GET /sys/reset

GET /sys/reset

系统重置

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) GET /sys/restore

GET /sys/restore

系统恢复出厂设置

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(四) GET /sys/device/get

GET /sys/device/get

获取设备名称

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{  
  "data": {  
    "device_name": "DECODER-D350159856666\n"  
  },  
  "result": "ok"  
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none

»» device_name	string	true	none		设备名称
» result	string	true	none		none

(五) POST /sys/device/set

POST /sys/device/set

设置设备名称

Body 请求参数

```
{
  "device_name": "D350-211111" //设备名称
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» device_name	body	string	是		设备名称

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /sys/ip/get

POST /sys/ip/get

根据 url 获取对应的 ip 端口

Body 请求参数

```
{
  "url": "ndi://10.20.30.21:5961" //url
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» url	body	string	是		url

返回示例

```
{
  "data": {
    "ip": "10.20.30.21",
    "path": "",
    "port": 5961
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» ip	string	true	none		ip
»» path	string	true	none		path
»» port	integer	true	none		端口
» result	string	true	none		none

(七) POST /sys/ip/replace

POST /sys/ip/replace

Body 请求参数

```
{
  "url": "srt://192.168.1.1:13?dddddd", //url
  "target_ip": "192.168.1.124", //目标 url
  "target_port": "1234" //目标端口
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

body	body	object	否		none
» url	body	string	是		url
» target_ip	body	string	是		目标 url
» target_port	body	string	是		目标端口

返回示例

```
{
  "data": {
    "url": "srt://192.168.1.124:1234?dddddd"
  },
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» url	string	true	none		替换后的 url
» result	string	true	none		none

十一、FIRMWARE 模块

(一) POST /firmware/upgrade

POST /firmware/upgrade

Body 请求参数

```
upload: string
path: D:\firmware-D350-kiloview-0.00.0000-20241010.bin
size: 文件长度
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
Content-Type	header	string	是		none
app	header	string	是		none
body	body	object	否		none
» upload	body	string(binary)	是		固件文件数据
» path	body	string	否		none

» size	body	integer	否		none
--------	------	---------	---	--	------

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

十二、PREVIEW 模块

(一) GET /preview/get

GET /preview/get

获取预览中的视频源列表

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": {
    "position": [
      {
        "height": 144,
        "id": 1,
        "resolution": "1920x1080p60",
        "status": 2,
        "stream_id": "e893d03a94326b981fd3aa276f90183a",
        "stream_name": "N60_238 (HB)",
        "stream_url": "192.168.36.238:5965",
        "type": "ndi",
        "width": 320,
        "x": 0,
        "y": 0
      }
    ]
  }
}
```

```

    },
    {
      "height": 144,
      "id": 2,
      "resolution": "640x360p30",
      "status": 2,
      "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a",
      "stream_name": "N60_238 (HX)",
      "stream_url": "192.168.36.238:5964",
      "type": "ndi",
      "width": 320,
      "x": 320,
      "y": 0
    }
  ]
},
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» position	\[object]	true	none		位置信息
»»» height	integer	true	none		高度
»»» id	integer	true	none		id
»»» resolution	string	true	none		分辨率信息
»»» status	integer	true	none		拉流状态
»»» stream_id	string	true	none		流 id
»»» stream_name	string	true	none		流名称
»»» stream_url	string	true	none		流 url
»»» type	string	true	none		流类型
»»» width	integer	true	none		显示宽度
»»» x	integer	true	none		图像位置

»»» y	integer	true	none		图像位置
» result	string	true	none		none

(二) POST /preview/source/modify

POST /preview/source/modify

添加视频源到预览中

Body 请求参数

```
{
  "from": {
    "type": "source", //来源信息
    "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a", //流 id
    "stream_name": "N60_238 (HX)", // //流名称
    "stream_url": "192.168.36.238:5964", //流 url
    "pos_id": "", //从某个位置过来
    "output_id": "1", //输出 id
    "layout_id": "" //布局 id
  },
  "to": {
    "type": "preview", //加入类型
    "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a", //流 id
    "stream_name": "N60_238 (HX)", //流名称
    "stream_url": "192.168.36.238:5964", //流 url
    "output_id": "1", //输出 id
    "layout_id": "" //布局 id
  }
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» from	body	object	是		none
»» type	body	string	是		来源信息
»» stream_id	body	string	是		流 id
»» stream_name	body	string	是		//流名称

»» stream_url	body	string	是		流 url
»» pos_id	body	string	是		从某个位置过来
»» output_id	body	string	是		输出 id
»» layout_id	body	string	是		布局 id
» to	body	object	是		none
»» type	body	string	是		加入类型
»» stream_id	body	string	是		流 id
»» stream_name	body	string	是		流名称
»» stream_url	body	string	是		流 url
»» output_id	body	string	是		输出 id
»» layout_id	body	string	是		布局 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) POST /preview/source/remove

POST /preview/source/remove

删除预览中的视频源

Body 请求参数

```
{
  "id": 1
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» pos_id	body	integer	是		位置 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

十三、NDI-Discovery 模块

(一) GET /ndi/discovery/all

GET /ndi/discovery/all

获取所有发现类型及配置

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none

返回示例

```
{
  "data": {
    "auto": [
      {
        "dis_d_ip": "",
        "dis_g_name": "123",
        "dis_method": 3,
        "dis_s_ip": ""
      }
    ],
    "discovery_server": [
      {
        "dis_d_ip": "",
        "dis_g_name": "public",
        "dis_method": 4,
        "dis_s_ip": "192.168.43.141"
      }
    ]
  }
}
```

```
},  
{  
  "dis_d_ip": "",  
  "dis_g_name": "puc",  
  "dis_method": 4,  
  "dis_s_ip": "192.168.43.141"  
}
```

```
],
```

```
"manual": [  
  {
```

```
    "dis_d_ip": "192.168.43.141",  
    "dis_g_name": "public",  
    "dis_method": 2,  
    "dis_s_ip": ""  
  },
```

```
  {  
    "dis_d_ip": "192.168.35.178",  
    "dis_g_name": "mhb",  
    "dis_method": 2,  
    "dis_s_ip": ""  
  },
```

```
  {  
    "dis_d_ip": "192.168.35.178",  
    "dis_g_name": "public",  
    "dis_method": 2,  
    "dis_s_ip": ""  
  }  
],
```

```
],
```

```
"public": [  
  {
```

```
    "dis_d_ip": "",  
    "dis_g_name": "public",  
    "dis_method": 1,  
    "dis_s_ip": ""  
  }  
]
```

```

},
"result": "ok"
}

```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none		none
»» auto_server	\[object]	true	none		自动发现服务器类型
»»» dis_d_ip	string	true	none		手动发现 ip
»»» dis_g_name	string	true	none		组名称
»»» dis_method	integer	true	none		发现方式 1:自动发现 2:手动发现 3:按组自动发现 4:服务器发现
»»» dis_s_ip	string	true	none		自动发现服务器 ip
»»» id	string	true	none		id 唯一值
»» manual	\[object]	true	none		手动列表
»»» dis_d_ip	string	true	none		none
»»» dis_g_name	string	true	none		none
»»» dis_method	integer	true	none		none
»»» dis_s_ip	string	true	none		none
»»» id	string	true	none		none
» result	string	true	none		none

(二) POST /ndi/discovery/auto/del

POST /ndi/discovery/auto/del

删除自动发现项

Body 请求参数

```
{
  "dis_d_ip": "192.168.35.93",
  "dis_g_name": "123"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	integer	是		唯一 id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(三) GET /ndi/discovery/sources

GET /ndi/discovery/sources

获取对应发现方式发现在源

Body 请求参数

```
{
  "dis_method": 1, //1:public 自动发现, 2:手动配置, 3:自动发现, 4:发现服务器发现
  "dis_g_name": "", //组名称, dis_method=3、4 时需要此参数
  "dis_s_ip": "", //发现服务器 IP, dis_method=4 时需要此参数
  "group_id": ""
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
dis_method	query	integer	否		1\:\:public 自动发现, 2:手动配置, 3:自动发现, 4:发现服务器发现

dis_g_name	query	string	否		组名称 , dis_method=3、 4 时需要此参数
dis_s_ip	query	string	否		发现服务器 IP , dis_method=4 时需要此参数
group_id	query	string	否		1
app	header	string	是		none
body	body	object	否		none

返回示例

```
{
  "data": [
    {
      "channel": "FULL",
      "group_name": "public",
      "ip": "192.168.35.178",
      "name": "N60-N6020240531 ",
      "ndi_name": "N60-N6020240531 (FULL)",
      "port": 5961,
      "url": "192.168.35.178:5961"
    }
  ],
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	\[object]	true	none		none
»» channel	string	false	none		通道名称
»» group_name	string	false	none		组名
»» ip	string	false	none		ip
»» name	string	false	none		设备名
»» ndi_name	string	false	none		NDI 发现的全名称

»» port	integer	false	none		端口
»» url	string	false	none		源 url
» result	string	true	none		none

(四) POST /ndi/discovery/manual/add

POST /ndi/discovery/manual/add

此接口也作为删除配置项接口用，用户在页面删除相应 ip,group 组后，将剩下的传给后台即可,跟矩阵的操作一样

Body 请求参数

```
{
  "manuals": [
    {
      "dis_d_ip": "192.168.43.141",
      "dis_g_name": "public"
    }
  ]
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» manuals	body	\[object]	是		none
»» dis_d_ip	body	string	是		发现 ip
»» dis_g_name	body	string	是		发现组名称

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(五) POST /ndi/discovery/auto/add

POST /ndi/discovery/auto/add

添加自动发现项

Body 请求参数

```
{
  "dis_g_name": "123"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» dis_g_name	body	string	是		组名称

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(六) POST /ndi/discovery/server/del

POST /ndi/discovery/server/del

删除服务器发现项

Body 请求参数

```
{
  "dis_s_ip": "192.168.43.146",
  "dis_g_name": "public"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» id	body	integer	是		id

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

(七) POST /ndi/discovery/server/add

POST /ndi/discovery/server/add

添加 server 发现项

Body 请求参数

```
{
  "dis_g_name": "public",
  "dis_s_ip": "192.168.43.141"
}
```

1. 请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是		none
body	body	object	否		none
» dis_g_name	body	string	是		组名称
» dis_s_ip	body	string	是		ip

返回示例

```
{
  "result": "ok"
}
```

2. 返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none		none

返回示例

```
{  
  "result": "ok"  
}
```